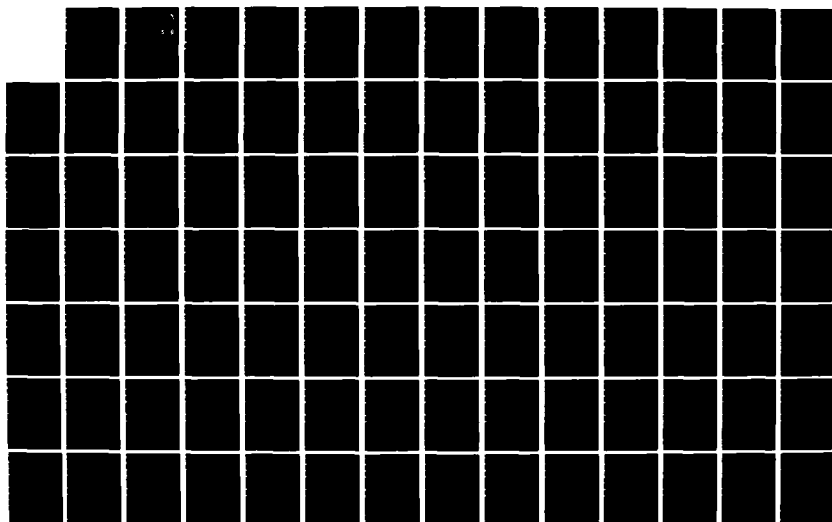


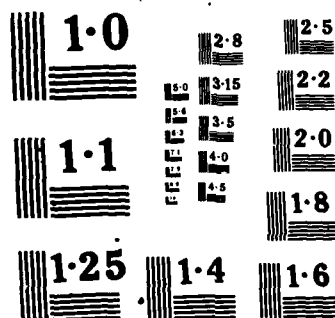
AD-A168 338 COMPUTER RESOURCES HANDBOOK FOR FLIGHT CRITICAL SYSTEMS 1/2  
(U) SOHAR INC LOS ANGELES CA H HECHT ET AL. JAN 85  
ASD-TR-85-5020 F33637-83-C-0103

UNCLASSIFIED

F/G 1/3

NL





NATIONAL BUREAU OF STANDARDS  
MICROCOPY RESOLUTION TEST

AD-A168 338

ASD-TR-85-5020

COMPUTER RESOURCES HANDBOOK FOR FLIGHT CRITICAL SYSTEMS

Herbert Hecht and Myron Hecht

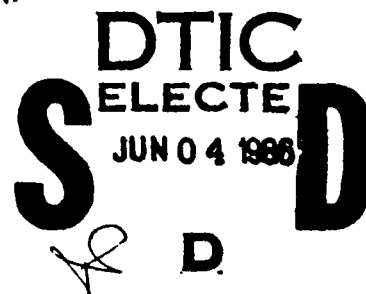
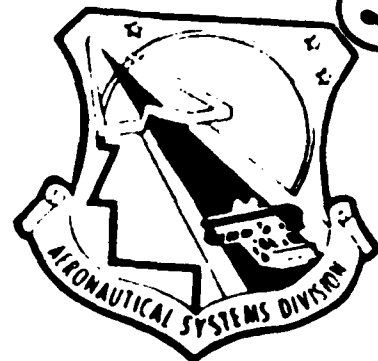
SoHaR Incorporated  
1040 S. LaJolla Avenue  
Los Angeles, CA 90035

January 1985

FINAL REPORT FOR PERIOD OCTOBER 1983 - SEPTEMBER 1985

Approved for public release; distribution unlimited.

DIRECTORATE OF FLIGHT SYSTEMS ENGINEERING  
AERONAUTICAL SYSTEMS DIVISION  
AIR FORCE SYSTEMS COMMAND  
WRIGHT-PATTERSON AIR FORCE BASE, OHIO 45433



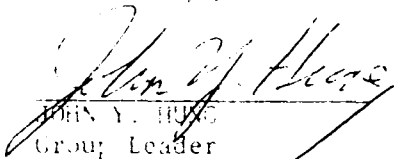
DTIC FILE COPY

NOTICE


When Government drawings, specifications, or other data are used for any purpose other than in connection with a definitely related Government procurement operation, the United States Government thereby incurs no responsibility nor any obligation whatsoever; and the fact that the government may have formulated, furnished, or in any way supplied the said drawings, specifications, or other data, is not to be regarded by implication or otherwise as in any manner licensing the holder or any other person or corporation, or conveying any rights or permission to manufacture, use, or sell any patented invention that may in any way be related thereto.

This report has been reviewed by the Office of Public Affairs (ASD/PA) and is releasable to the National Technical Information Service (NTIS). At NTIS, it will be available to the general public, including foreign nations.

This technical report has been reviewed and is approved for publication.

  
JOHN Y. HINES  
Group Leader  
Computer Resources  
Flight Systems Engineering

FOR THE COMMANDER

  
JAMES W. HOLLER, III, Col, USAF  
Director  
Flight Systems Engineering

If your address has changed, if you wish to be removed from our mailing list, or if the addressee is no longer employed by your organization please notify ASD/ENF2, WPAFB OH 45433, to help us maintain a current mailing list.

Copies of this report should not be returned unless return is required by security considerations, contractual obligations, or notice on a specific document.



UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE

## REPORT DOCUMENTATION PAGE

1a. REPORT SECURITY CLASSIFICATION UNCLASSIFIED			1d. RESTRICTIVE MARKINGS										
2a. SECURITY CLASSIFICATION AUTHORITY			3. DISTRIBUTION/AVAILABILITY OF REPORT										
2b. DECLASSIFICATION/DOWNGRADING SCHEDULE			Approved for public release; distribution unlimited										
4. PERFORMING ORGANIZATION REPORT NUMBER(S)			5. MONITORING ORGANIZATION REPORT NUMBER(S)										
			ASD-TR-85-5020										
6a. NAME OF PERFORMING ORGANIZATION		6b. OFFICE SYMBOL (If applicable)		7a. NAME OF MONITORING ORGANIZATION									
SoHaR Incorporated				Dir of Flight Systems Engineering Aeronautical Systems Division									
6c. ADDRESS (City, State and ZIP Code)			7b. ADDRESS (City, State and ZIP Code)										
SoHaR Incorporated 1040 S. LaJolla Ave. Los Angeles CA 90035			ASD/ENFZ Wright-Patterson AFB OH 45433										
8a. NAME OF FUNDING/SPONSORING ORGANIZATION		8b. OFFICE SYMBOL (If applicable)		9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER									
				F33657-83-C-0183									
8c. ADDRESS (City, State and ZIP Code)			10. SOURCE OF FUNDING NOS.										
			<table border="1"> <tr> <td>PROGRAM ELEMENT NO.</td> <td>PROJECT NO.</td> <td>TASK NO.</td> <td>WORK UNIT NO.</td> </tr> <tr> <td>PE 64740F</td> <td>2524</td> <td></td> <td></td> </tr> </table>			PROGRAM ELEMENT NO.	PROJECT NO.	TASK NO.	WORK UNIT NO.	PE 64740F	2524		
PROGRAM ELEMENT NO.	PROJECT NO.	TASK NO.	WORK UNIT NO.										
PE 64740F	2524												
11. TITLE (Include Security Classification)													
Computer Resources Handbook for Flight Critical Systems													
12. PERSONAL AUTHOR(S) Herbert Hecht and Myron Hecht													
13a. TYPE OF REPORT		13b. TIME COVERED		14. DATE OF REPORT (Yr., Mo., Day)									
Final		FROM 10/83 to 9/85		January 1985									
15. PAGE COUNT 177													
16. SUPPLEMENTARY NOTATION													
17. COSATI CODES			18. SUBJECT TERMS (Continue on reverse if necessary and identify by block number)										
FIELD	GROUP	SUB. GR.	Reliability and Fault Tolerance (RAFT)										
01	03		Computer Resources										
			Flight Critical Systems										
19. ABSTRACT (Continue on reverse if necessary and identify by block number)													
<p>The functional capabilities of digital devices together with their comparatively low cost and physical resource requirements make it desirable to use computerbased systems in all areas of USAF activities and particularly for those aboard aircraft. Of special concern is the use of such systems where the failure of a computer can cause loss of the aircraft and flight crew -- the use of computers in flight critical applications. Special reliability and fault tolerance (RAFT) techniques are being used within ASD and also in other military and civilian aircraft organizations to minimize and cope with the effect of failure. However, each installation of computers in connection with a flight critical function is being treated as a special case, and there are few guidelines for establishing requirements for such systems, managing their development or conducting acceptance or certification tests. This Computer Resources Handbook for Flight Critical Systems is intended as a step in filling this need. The Handbook is intended to cover the entire life cycle of a weapon system: concept definition, development, test, and operation and</p>													
20. DISTRIBUTION/AVAILABILITY OF ABSTRACT			21. ABSTRACT SECURITY CLASSIFICATION										
UNCLASSIFIED/UNLIMITED <input checked="" type="checkbox"/> SAME AS RPT. <input type="checkbox"/> DTIC USERS <input type="checkbox"/>			UNCLASSIFIED										
22a. NAME OF RESPONSIBLE INDIVIDUAL			22b. TELEPHONE NUMBER (Include Area Code)		22c. OFFICE SYMBOL								
JOHN Y. HUNG			(513) 255-3880		ASD/ENFZ								

DD FORM 1473, 83 APR

EDITION OF 1 JAN 73 IS OBSOLETE.

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE

maintenance. Emphasis is placed on the early stages of the life cycle because deficiencies introduced there can be remedied in later stages only at very great cost.

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE

# TABLE OF CONTENTS

SECTION	SUBJECT	PAGE
1.	INTRODUCTION . . . . .	1
1.1	Objectives of this Effort. . . . .	1
1.2	Methodology Used . . . . .	2
1.3	Organization of the Handbook . . . . .	2
1.4	Acknowledgements . . . . .	4
2.	THE ROLE OF THE CONTRACTING ORGANIZATION . . . . .	5
2.1	Establishing Requirements for Flight Critical Systems. . . . .	5
2.2	Identification of Flight Critical Systems. . . . .	7
2.3	Contractor Organization for Product Assurance. . . . .	8
2.4	Independent Verification and Validation. . . . .	10
2.5	Compliance with Reliability and Fault Tolerance Requirements. . . . .	11
3.	FAILURES IN DIGITAL EQUIPMENT. . . . .	15
3.1	Classification of Failures . . . . .	15
3.2	Experience on Current Systems. . . . .	19
3.3	Extrapolation to Future Systems. . . . .	20
3.4	A Unified Model for Failures in Digital Systems. . . . .	22
4.	DEFINITION OF FLIGHT CRITICAL SYSTEMS. . . . .	29
4.1	Criteria for Criticality . . . . .	29
4.2	Criticality by Aircraft Type and Mission Phase . . . . .	32
4.3	Criticality by Aircraft System . . . . .	37
4.3.1	Direct Access Function . . . . .	37
4.3.2	Indirect Access Functions. . . . .	37
4.3.3	Human Mediated Functions . . . . .	41
4.3.4	Non-Control Functions. . . . .	42
4.4	Criticality of System Interfaces . . . . .	42
4.4.1	Utility Interfaces . . . . .	42
4.4.2	Interfaces on Dedicated Links. . . . .	46
4.4.3	Bus Interfaces . . . . .	46
4.4.4	Software Interfaces. . . . .	49
4.5	Criticality of Sensor and Actuator Interfaces. . . . .	51
4.5.1	Detection of Sensor and Actuator Failures. . . . .	51
4.5.2	Response to Actuator and Sensor Failures . . . . .	52
4.6	Problems in System Integration . . . . .	53



Distribution /	
Availability Codes	
Dist	Avail and/or Special
A-1	

SECTION	SUBJECT	PAGE
5.	TECHNIQUES OF RELIABILITY, FAULT CONTAINMENT AND FAULT TOLERANCE	57
5.1	Conventional Reliability Improvement Techniques (Fault Avoidance) . . . . .	57
5.2	Systems Oriented Reliability Improvement . . . . .	60
5.2.1	Failure Mode - Effects and Criticality Analysis . . . . .	60
5.2.2	Sneak Circuit Analysis . . . . .	64
5.2.3	Fault Tree Analysis . . . . .	66
5.2.4	Dynamic Analysis of Computer Programs . . . . .	69
5.3	Fault Containment . . . . .	70
5.4	Hardware Fault Tolerance - Codes and Repetition . . . . .	74
5.4.1	Cyclic Codes . . . . .	75
5.4.2	Array Codes . . . . .	77
5.4.3	Repetition . . . . .	78
5.5	Hardware Fault Tolerance - Redundancy . . . . .	79
5.5.1	Static and Dynamic Fault Tolerance . . . . .	79
5.5.2	Configurations Employing Two Computers . . . . .	80
5.5.3	Configurations Employing Three Computers . . . . .	81
5.5.4	Configurations Employing Four Computers . . . . .	82
5.5.5	Multiprocessor Systems . . . . .	84
5.6	Fault Tolerant Software . . . . .	86
5.6.1	N-Version Programming . . . . .	89
5.6.2	Recovery Block Programming . . . . .	89
5.7	Aircraft Level Fault Tolerance . . . . .	91
5.7.1	Distributed Computing . . . . .	93
5.7.2	Central Malfunction and Damage Control . . . . .	93
5.7.3	Use of Alternate Control Modes . . . . .	94
5.7.4	Lifeboat Systems . . . . .	94
5.8	General Application Notes for Fault Tolerance . . . . .	95
5.8.1	Partitioning for Fault Tolerance . . . . .	95
5.8.2	Similar vs. Dissimilar Redundancy . . . . .	99
5.8.3	Response Time Requirement for Recovery from a Failure . . . . .	100
5.8.4	Integration of Fault Tolerance and Diagnostic Capabilities . . . . .	101
5.9	Incorporation of Safety Objectives . . . . .	102
5.10	Summary and Trade-Off Criteria . . . . .	105
6.	EVALUATION METHODOLOGY . . . . .	107
6.1	Evaluation Criteria . . . . .	107
6.1.2	Utilization of Criteria . . . . .	109
6.2	Analytical Models . . . . .	110
6.2.1	Simple Analytical Models . . . . .	110
6.2.2	Modifications of Simple Models . . . . .	111
6.3	Simulations . . . . .	112
6.4	Reliability and Fault Tolerance Evaluation During Development . . . . .	115
6.5	Reliability and Fault Tolerance Evaluation During Test . . . . .	118
6.6	Reliability and Fault Tolerance Evaluation During Operation . . . . .	119

SECTION	SUBJECT	PAGE
7.	APPLICATION OF RELIABILITY AND FAULT TOLERANCE . . . . .	122
7.1	RAFT Requirements for the Concept Definition Phase . . . . .	122
7.1.1	Provisions for the System Specification. . . . .	122
7.1.2	Studies and Activities Requirements. . . . .	124
7.1.3	Verification Provisions. . . . .	126
7.2	RAFT Requirements for the Development Phase. . . . .	126
7.2.1	Provisions for the System Specification. . . . .	126
7.2.2	Studies and Activities Requirements. . . . .	128
7.2.3	Verification Provisions. . . . .	129
7.3	REQUIREMENTS FOR A RELIABILITY IMPROVEMENT PROGRAM . . . . .	131
7.3.1	Provisions for the System Specification. . . . .	131
7.3.2	Studies and Activities Requirements. . . . .	132
7.3.3	Verification Provisions. . . . .	133
	REFERENCES . . . . .	134
APP. A	GLOSSARY . . . . .	139
A.1	Abbreviations and Acronyms . . . . .	139
A.2	Full Nomenclature of Government Documents. . . . .	140
APP. B	EXCERPTS FROM FEDERAL AVIATION REGULATIONS . . . . .	143
APP. C	BIBLIOGRAPHY . . . . .	147
C.1	General. . . . .	147
C.2	Specific Applications. . . . .	148
C.3	Specific Techniques. . . . .	148
APP. D	EXPERIENCE WITH FLIGHT CRITICAL SYSTEMS. . . . .	151
D.1	Production Installations . . . . .	151
D.2	Experiemental Installations. . . . .	153
APP. E	AIRCRAFT ELECTRIC POWER SYSTEMS. . . . .	161
E.1	Availability of Power during Normal Flight Conditions. . . . .	162
E.2	Power Conversion . . . . .	168
E.3	Non-Flight and Emergency Conditions. . . . .	169
E.4	Built-In Test (BIT). . . . .	170
E.5	Control of Electromagnetic Interference (EMI) . . . . .	172

# LIST OF FIGURES

FIG. NO.	TITLE	PAGE
2-1	Example of Total System V&V. . . . .	12
3-1	Conventional Reliability Diagram . . . . .	24
3-2	Load Strength Model. . . . .	24
3-3	Basic Failure Model. . . . .	25
3-4	Modeling a Pitch Axis Failure. . . . .	25
3-5	Model Applied to a Fault Tolerant System . . . . .	27
3-6	Effect of Input/Output Operations. . . . .	27
4-1	AC Voltage Limits from MIL-STD-704D. . . . .	44
4-2	DC Voltage Limits from MIL-STD-704D. . . . .	45
4-3	Generation of Actuator Position Error Signal . . . . .	54
5-1	Economically Optimal Reliability . . . . .	59
5-2	FMECA Worksheets . . . . .	62
5-3	Example of Criticality Matrix. . . . .	63
5-4	Example of a Sneak Circuit . . . . .	65
5-5	Example of a Fault Tree. . . . .	67
5-6	Example of a Software Fault Tree . . . . .	68
5-7	Dynamic Analysis Procedure. . . . .	71
5-8	Dynamic Analysis Reports . . . . .	72
5-9	General Organization of the FTMP . . . . .	85
5-10	General Organization of SIFT . . . . .	87
5-11	Architecture and Operation of the CRMMP. . . . .	88
5-12	Example of Real Time Recovery Block. . . . .	90
5-13	Recovery Block and System Executive. . . . .	92
5-14	Effect of Partitioning on Reliability. . . . .	96
5-15	Fast Response for Reconfiguration of External Computers. . . . .	100
5-16	Effect of Extreme Position of Low Pressure Turbine Inlet Control. . . . .	103
6-1	General Form of Parallel Redundancy. . . . .	110
6-2	Example of State Transition Model. . . . .	113
6-3	Transition Model Requiring Simulation. . . . .	114
6-4	Typical Transition Elements of Care III. . . . .	116
D-1	F-16 Digital Flight Control System . . . . .	154
D-2	F-16 Flight Control Computer . . . . .	155
D-3	Servo Actuator Interfaces of the F-16 DFCS . . . . .	156
D-4	Leading Flap Interface of the F-16 . . . . .	157
D-5	AFTI F-16 Digital Flight Control System. . . . .	159
E-1	Typical Single Generator Installation. . . . .	164
E-2	Typical Two Generator Installation. . . . .	165
E-3	Differential Current and Zone Protection . . . . .	167
E-4	Emergency Load Transfer. . . . .	171

# LIST OF TABLES

TBL. NO.	TABLE	PAGE
3-1	Classification of Causes of Failure . . . . .	17
3-2	Civil Aircraft Accidents/Incidents Involving Flight Controls (1972-81) . . . . .	19
3-3	Effect of Language on Fault Density . . . . .	22
4-1	Checklist for Selection of Flight Critical Functions. . . . .	31
4-2	Typical Flight-Critical Equipment for Trainers. . . . .	33
4-3	Typical Flight-Critical Equipment for Transports. . . . .	33
4-4	Typical Flight-Critical Equipment for Surveillance Aircraft. . . . .	34
4-5	Typical Flight-Critical Equipment for Fighter Aircraft. . . . .	35
4-6	Typical Flight-Critical Equipment for Bombers . . . . .	36
4-7	Typical Flight-Critical Equipment for Helicopters . . . . .	36
4-8	Typical Flight-Critical Equipment for Unmanned Aircraft . . . . .	37
5-1	Example of Cyclic Error Correcting Code . . . . .	76
5-2	Array Codes . . . . .	78
5-3	Fault Tolerant Computer Configurations. . . . .	83
5-4	Potential System Support Impact on Partitioning . . . . .	98
5-5	System Benefit Evaluation . . . . .	106
D-1	Production Digital Flight Control Installations . . . . .	152
D-2	USAF Aircraft Incidents Involving Digital Systems . . . . .	152
D-3	Natural Faults in F-8 DFBW Flights. . . . .	158

## Chapter 1

### INTRODUCTION

#### 1.1 OBJECTIVES OF THIS EFFORT

The functional capabilities of digital devices together with their comparatively low cost and physical resource requirements make it desirable to use computer based systems in all areas of USAF activities and particularly for those aboard aircraft. Of special concern is the use of such systems where the failure of a computer can cause loss of the aircraft and flight crew -- the use of computers in flight critical applications. Special reliability and fault tolerance (RAFT) techniques are being used within ASD and also in other military and civilian aircraft organizations to minimize and cope with the effects of failure. However, each installation of computers in connection with a flight critical function is being treated as a special case, and there are few guidelines for establishing requirements for such systems, managing their development or conducting acceptance or certification tests. This Computer Resources Handbook for Flight Critical Systems is intended as a step in filling this need.

The Statement of Work identifies the objectives for this effort as:

1. to define what is flight critical in terms of components and configuration;
2. to explain how to specify requirements for flight critical systems; and
3. to tell the handbook users how to evaluate contractor's proposed designs.

The Handbook is intended to cover the entire life cycle of a weapon system: concept definition, development, test, and operation and maintenance. Emphasis is placed on the early stages of the life cycle because deficiencies introduced there can be remedied in later stages only at very great cost.

The primary users of this Handbook are ASD project engineers in charge of flight critical systems. It is assumed that the reader has a general engineering background, is familiar with the functional requirements of the application (flight control, engine control, etc.), and will utilize other guidance documents for the administrative aspects of his job (e. g., AFR 800-14 for acquisition and support procedures). The Handbook introduces the reader to the terminology and basic concepts of RAFT techniques, establishes definitions of critical systems, and provides a general methodology as well as specific examples for the application of RAFT techniques in the development and operation of a weapon system. A brief review of the content is presented in 1.3.

The selection of a specific technique is seldom governed exclusively by technical factors, such as computer performance or expected reliability. The selection of a system based on prior use, availability of logistic support (test



equipment, documentation, spares), or considerations of national policy must be anticipated. Therefore the evaluation is frequently presented in terms of acceptable solutions rather than optimal ones. Also, general principles of economic trade-off procedures are included.

## 1.2 METHODOLOGY USED

To permit the reader to judge the usefulness of this volume and of individual sections of it in his or her environment a brief overview of the methodology is presented here. It is assumed that any system procured for aircraft use by ASD will utilize high quality parts, will be assembled in accordance with the best practice, and will undergo testing under Government supervision. Therefore many failure mechanisms that are present in consumer goods will have been eliminated. These procedures still leave a probability of failure that is unacceptably high for flight critical use. Thus, a requirement for fault tolerance is implicit once a component or system is shown to be essential for safe continuation of flight or landing. A substantial portion of the Handbook therefore deals with the definition of criticality in general terms and with the identification of flight critical systems and components.

Many reported aircraft accidents are due to multiple causes, such as a poorly designed access door together with faulty maintenance which permitted the aircraft to take off without engaging all fastening devices for the door. Similarly, the Three Mile Island accident involved compounded effects of five or six individually non-critical deficiencies. It is recognized that aircraft and nuclear power stations involve safety risks, and precautions are therefore taken to avoid or circumvent all single causes of failure that could lead to a catastrophic event. But it is also necessary to consider multiple causes and these are included in the methodology presented here.

Failures in many components in common use are either due to a recognized wear-out process (e. g., an automobile battery or tire) or are permanent failures that are attributed to random causes (e. g., a transistor in a radio or television set). Failures in digital equipment are frequently of a transient nature which makes diagnosis very difficult; they are also likely to be design related rather than due to random causes. Software failures always represent a design fault although they may manifest themselves in a random manner because they are triggered by a specific data set or computer state. Conventional diagnosis is inadequate for dealing with transient failures, and conventional redundancy is inadequate for dealing with design faults. Therefore flight critical applications frequently need multiple fault tolerance provisions, and the design and evaluation of these are specifically covered in this Handbook.

## 1.3 ORGANIZATION OF THE HANDBOOK

Flight critical systems are a part of the aircraft, and therefore many aspects of their development and test must be left to the discretion of the airframe contractor. On the other hand, the Government generates the requirements for the aircraft, decides on the missions and operational doctrine, and is concerned with the support of the systems throughout their service life. Thus, there is a

division of responsibilities between the sponsor of a development project and the contractor, and the next chapter of this Handbook identifies the areas of concern of the contracting organization. These include not only the form and function of the final product but also the process attributes. The organizational aspects of product assurance and of verification and validation are described in this connection, and the need for careful formulation of compliance provisions is discussed.

In 1.2 it was briefly mentioned that failures in digital equipment frequently manifest themselves in ways that differ from those encountered in most other devices. A detailed discussion of failure modes in digital components is therefore provided in Chapter 3 of the Handbook. The high incidence of transient and intermittent failures is emphasized, as is the need to protect against failures induced by the environment or propagated across interfaces. A unified hardware/software failure model is introduced that views failure as due to the interaction of an inherent fault or weakness with events in the environment and that accounts for observation of failures at several levels.

Chapter 4 of the Handbook covers the definition of flight critical systems and components. Criteria for criticality are discussed first, and this is followed by an assessment of critical components by aircraft type, mission phase, and by a description of the subsystems in which critical components are most likely to be found. Criticality considerations arising at system interfaces are emphasized, and special practices for sensor and actuator interfaces are presented. Functional and performance benefits frequently motivate the integration of flight critical systems with weapon control and related functions, and the issues that arise from this integration are therefore evaluated. Chapter 4 represents the transition from broad concerns with the management of flight critical systems to the practice of failure prevention at the technical level.

The next chapter deals with specific techniques for reliability, fault containment and fault tolerance in flight critical systems. Conventional reliability techniques and analysis and reliability improvement techniques at the system level are described. Because temporary failures are prominent in digital equipment, specific techniques for dealing with these, primarily fault containment, are discussed in a separate section. Other hardware and software fault tolerance techniques are presented together with an evaluation of their effectiveness and relative cost. A methodology for trade-offs in the fault tolerance area concludes that chapter.

A key responsibility of the contracting organization is to evaluate the attainment of fault avoidance and fault tolerance throughout the development and into the operational phase. This subject is covered in Chapter 6 of the Handbook. Evaluation criteria, analytical models and simulation models are described. The conduct of RAFT evaluation during development and test are discussed, the problems of continuing the evaluation in the operational phase are outlined, and some measures of dealing with these problems are presented.

The final Chapter of the body of this document presents examples of the application of these techniques. The first example is concerned with the statement of requirements for a flight control system during the concept definition phase, the second one with RAFT implementation during the development of a turbojet engine, and the third with the conduct of a reliability improvement program after an aircraft has entered the operational phase.

The appendices form an important part of this Handbook. Appendix A contains an explanation of abbreviations and acronyms and the complete nomenclature of DoD standards and specifications utilized in the preparation of this document. These are referred to in the body of the Handbook without revision letter when the generic content of the document is involved; when specific paragraphs or provisions are cited the revision letter is shown. Excerpts from provisions of the Federal Aviation Regulations that cover flight critical systems in civil aircraft are presented in Appendix B. A bibliography is provided in Appendix C, and source data on the reliability of flight critical systems are summarized in Appendix D. Design notes for electric power systems are provided in Appendix E.

#### 1.4 ACKNOWLEDGEMENTS

The creation of this Handbook has been undertaken at the request of the Computer Resources Group in the Flight Systems Engineering Directorate of the Aeronautical Systems Division. Technical Direction was provided by Mr. John Y. Hung who has made many constructive suggestions for the format and content of this volume. The research leading to the findings presented here benefited from discussions with many individuals in ASD and in other Air Force organizations including:

Mr. Paul Adams, AFWAL/POTC  
Mr. Evard Flinn, AFWAL/FIGL  
Mr. Charles Gross, AFALC/MMEA  
Lt. Col. George Meinke, ASD/(B1 Project Office)  
Mr. Carroll Wiedenhouse, AFALC/PTR

Personnel in other Government organizations provided materials that were included in this volume or constituted valuable background information for the materials presented here, including:

Dr. Bernard Loeb, National Transportation Safety Board  
Mr. J. B. McCollough, FAA Technical Center  
Mr. Richard Larson, NASA Dryden Flight Research Facility  
Mr. Dale A. Mackall, NASA Dryden Flight Research Facility  
Mr. Francis Rock, FAA Office of Airworthiness

Significant help was received from private organizations in the aircraft and avionics fields, particularly:

Mr. Dennis Mulcare, Lockheed Georgia Company  
Mr. S. S. Osder, Sperry Flight Systems Division

The author wishes to express sincere thanks to all of these and also want to emphasize that none of these individuals is responsible for errors that might have arisen in presenting or interpreting information furnished to us that is included in this report.

## Chapter 2

### THE ROLE OF THE CONTRACTING ORGANIZATION

This chapter describes the framework within which decisions on flight critical systems and components are made in the typical DoD development program. While personnel representing the contracting organization usually have no direct design responsibility, there are many ways in which the statement and format of the requirements and of the program plan can affect the design. The purpose of the first two sections within this chapter is to surface these effects with particular emphasis on flight critical systems. This is followed by a discussion of outside organizations which can be of help in the management of flight critical items: the developer's product assurance organization and an independent verification and validation organization. The final section in this chapter establishes the need for compliance provisions and outlines a suitable format for these.

The procurement of computer resources is governed by a number of DoD and departmental regulations, such as AFR 800-14. The responsibilities and actions discussed below are believed to be consistent with these regulations as of the writing of this report. However, because of the changing nature of acquisition regulations the users of this Handbook are cautioned to ascertain the specific constraints applicable to their project, and to tailor the guidance provided below to fit within these constraints.

#### 2.1 ESTABLISHING REQUIREMENTS FOR FLIGHT CRITICAL SYSTEMS

Prior to discussing specific criticality designations it is desirable to understand the consequences that may arise from these designations. Declaring a component of a system as critical may result in one or more of the following actions:

- Modifications of system structure or function;
- Special handling of the designated component;
- Training of air crews to deal with consequences of a failure; or
- Restriction of the operational envelope.

The last of these is the least acceptable one for Air Force applications (as well as for most others). The acceptability of that option is based on operational factors and is not further discussed within this chapter. Requirements for special training are also undesirable because (a) training of air crews already occupies a large fraction of the total period of crew time in the military services, and any additional training will reduce the availability

for operational missions, (b) training for dealing with exceptional events must be reinforced by frequent in-service check flights since the performance under these conditions is not enhanced or observed during routine missions, and (c) even intensive training cannot ensure that the crew will react appropriately to an in-flight emergency which may not exactly duplicate the training situation. This leaves the first two courses of action as the ones that are of primary concern in this Handbook.

The most common modification of system structure to deal with critical components is to employ redundancy. This can take the form of two identical components (e. g., two hydraulic pumps of the same design) or of two deliberately different components serving the a given function, e. g., a magnetic and a gyroscopic compass. An example of the modification of system function is to use direct angle of attack measurement for a stall warning system rather than to derive the warning from quantities which are already displayed to the pilot. All of these measures increase the direct cost of the aircraft and impose indirect and upkeep costs through the weight, power, documentation and maintenance requirements of the added equipment.

Special handling typically involves advanced quality assurance techniques during manufacture and application. Such techniques are particularly applicable to components which have a single failure mode that is well understood so that signs of potential failure can be detected during manufacturing, installation, and as part of maintenance. Many structural and mechanical aircraft parts fall into this category. For digital equipment special handling techniques are primarily used to reduce the probability of failure rather than as an absolute failure prevention measure. Special handling provisions may consist of specifying a maximum failure rate for a component as a whole, defining the procedure by which reliability shall be demonstrated, or requiring that parts and processes comply with high quality levels in an applicable specification. Such measures can supplement other means of dealing with component criticality.

It is seen that designation of a system or of a component as critical carries with it a commitment to resource expenditures (beyond those required for the function proper) for design, procurement, test, and maintenance activities. A major portion of this Handbook (particularly Chapters 4 - 6) provides information for trade-offs to arrive at economically optimal measures for dealing with requirements arising from criticality designations. However, economies at the detail level can be insignificant relative to savings that are possible by care in the overall system concept that can either avoid critical components altogether or restrict them to partitions that have minimal interactions with non-critical portions. It is quite obvious that overclassification of the criticality of a component will result in unnecessary expenditures and should therefore be avoided.

As will be seen in Chapter 3, it is impossible or, at the least, very difficult to prevent failures in non-critical portions of a computer from affecting operations in critical portions. Therefore it is generally desirable to segregate information processing tasks based on the level of criticality of the function served by them. This rule may be modified where the means for dealing with the highest criticality level are already provided (e. g., in terms of a fault tolerant computer).

These details are provided as background for the role of the contracting organization rather than as a basis for specific actions. In most cases it is desired to state requirements for a weapons system (or for portions of it)

without implying or requiring a specific design. A methodology for accomplishing this is outlined in the following heading.

## 2.2 IDENTIFICATION OF FLIGHT CRITICAL SYSTEMS

Under this heading the procedural and management aspects of the identification of flight critical systems are discussed. The criticality classification and criteria are the subject of Chapter 4 of this Handbook.

The purpose of identifying certain digital subsystems and components as critical is to alert the development team, the user, and the maintenance organization to the fact that failure of these items will have adverse effects on the crew, aircraft or mission. The exact nature of these effects is usually unknown at the time the systems requirements for an aircraft are generated but guidelines are available in military specifications and in the certification procedures for commercial aircraft (the latter will have to be interpreted for military applications). Criticality may be specified in one of the following forms:

1. By reference to MIL-F-9490(for flight control systems);
2. By reference, with tailoring if required, to a Federal Aviation Regulation (FAR). Pertinent FAR excerpts are reproduced in Appendix B;
3. By declaring an item to be critical for a function or segment (which may be a flight or mission phase or the entire flight);
4. By reference to a subsystem performance specification (which may identify a criticality level or may state that the system must tolerate at least a specified number of failures);
5. By stating that a minimum level of service must be maintained after specified failure conditions; or
6. By requiring suitably high reliability or availability for a function, together with test or demonstration procedures.

The last three forms imply rather than state criticality. The translation from criticality to a fault tolerance or reliability requirement has already been accomplished in those cases. While these implicit identifications of the criticality of a subsystem or function represent the current state of practice and, in some cases, have a legal standing, their adoption is not always advisable because (i) preventive measures may be adopted without adequate analysis of the criticality in a given application, (ii) inconsistent protection may be selected for functions of the same criticality, and (iii) advances in technology may offer more suitable methods of analysis or failure circumvention than are specified in existing documents.

The distinction between mission critical and flight critical functions sometimes becomes blurred in weapon platforms. Failure of flight or engine control functions that are required only for extreme maneuvers can cause loss of crew and aircraft under combat conditions whereas unavailability of these functions in other situations can be tolerated. Consideration should be given to recognize this special class by a designation "flight critical -- combat". The

provisions for dealing with this criticality class will depend on how closely the special functions are tied to the general flight and engine control functions, the magnitude of the performance increment due to the special functions, and the resources required for providing various levels of fault tolerance.

To prevent errors or anomalous conditions from propagating to flight critical functions, interfaces to flight critical systems and components must receive special management attention to:

- keep their number and complexity within bounds;
- maintain functional autonomy on each side of the interface; and
- exercise strict configuration control over each interface.

Functional autonomy is important in order to facilitate test and validation of the critical functions as a separate entity and in order to reduce the number of changes that affect information flowing across the interface. These considerations reinforce what has already been said in the previous heading about the importance of keeping criticality considerations in mind in the partitioning of the aircraft and its subsystems.

Critical subsystems and components require special attention not only during the development phase but throughout the service life of the aircraft. Incidents affecting critical items are subject to mandatory reporting; maintenance on critical items frequently has to be carried out by certified personnel; and engineering changes may require revalidation of the item or of substantial portions of it. The incremental cost of these activities must be factored into the decisions made during the development or major upgrade of an aircraft.

### 2.3 CONTRACTOR ORGANIZATION FOR PRODUCT ASSURANCE IN CRITICAL SYSTEMS

Organizational independence of the contractor's product assurance group from the development group is recognized as an important factor in ensuring a thorough review of critical items. Organizational independence permits challenging of assumptions, terminology, and product characteristics that originated in the development area. Also, an independent product assurance group knows that its reputation (and future business) depends on identifying all discrepancies in the submitted articles. It is therefore motivated not to overlook faults.

Organizational independence can take several forms. The product assurance group may be part of the company that is responsible for the development but report to a level of management that has much broader responsibilities than the specific development that is being examined. In this way it is expected that concern for the integrity and reputation of the company will outweigh budgetary and schedule concerns of the development program. This arrangement is typical with regard to individual portions of a hardware or software development. For the overall performance of the aircraft or of a major subsystem a greater degree of organizational independence is usually required, and this can only be met by contracting for this work to be performed by a different company. The independent verification and validation discussed in 2.4 is a special case of such an effort directed at computers and software.

The product assurance process may identify some deficiencies or inconsistencies which do not in any way compromise the functions to be performed by the items that have been examined. The decision to waive corrective action in these instances must be referred to a review board that includes personnel of both the contracting and the contractor organization. It is essential that personnel assigned to that board have detailed technical knowledge and managerial background. The technical knowledge must cover not only the function of the item in question but also the overall aircraft performance requirements. Where this combined capability is not available from a single individual, several representatives may be named. The managerial background is required to comprehend the impact of the decisions on the contractual obligations of the parties, and particularly to assess whether a waiver granted with regard to design detail or to attributes of a single part can be interpreted to imply or constitute a waiver of overall system performance or reliability requirements.

The Military Standard for Technical Reviews and Audits for Systems, Equipments and Computer Programs, MIL-STD-1521A, does not provide specific guidance for the review of flight critical components or systems. However, there are provisions for system safety subjects in each of the reviews (e. g., par 30.8 for the Preliminary Design Review), and critical items can be covered within that framework. As in the case of the review boards mentioned above, it is important that the contracting organization be represented by personnel who have both broad background in aircraft development and operation and specific capabilities in the components under review.

Design, product assurance, and manufacturing personnel may require special training to understand:

- the importance of their assignment to the overall mission;
- why the component or system is flight or mission critical;
- specific technical and contractual requirements governing their work; and
- procedures for reporting non-compliance.

Evidence of comprehension and retention of these subjects may be obtained by oral or written examinations. Attendance at the training sessions and satisfactory performance in the examination may form the basis for certification of personnel who are in responsible positions in the development, manufacture, test, or maintenance of critical items.

Such training is expensive, and it can be given only to a select group of participants in the program. Abbreviated versions of the material may be used for training personnel with more restricted responsibilities, and some may not receive any formal training at all. It is important to communicate the flight or mission critical nature of an item to all who handle the article itself or documents (including drawings, test reports, etc.) related to it so that supervisors can assign trained personnel to perform the work or provide required guidance to untrained personnel. Marking of documents and, where appropriate, of the item itself should be required in order to facilitate proper handling.

Problems have frequently been encountered in communicating to subcontractors the critical nature of items in which they are involved. Subcontractors are usually much smaller organizations, and, although they are expert in a specialty area,



management may not understand the end use to which their product will be put in a weapon system. The responsibility on the prime contractor to communicate reliability and safety requirements to the subcontractors and to verify their compliance is spelled out in military standards (e. g., MIL-STD-785A, par. 5.1.3, and MIL-STD-882A, par. 5.1.2d). In addition to adherence to these requirements it may be desirable to include subcontractors in training programs, or to establish separate ones for them, specifically tailored to their environment and responsibilities. Subcontractors should also be required to attend design, reliability, and safety reviews and to participate in review boards when the activities of these are concerned with subcontracted items.

The hardware/software interface needs special attention in the management of flight critical items. Frequently software is depended on to diagnose and recover from hardware failures. The proper execution of these extremely sensitive programs requires close collaboration between hardware and software experts and must be supervised by an individual who has adequate background in both areas. Testing of diagnostic and recovery routines must consider all combinations of operating modes and timing under which the failures may occur and adequate resources must therefore be allocated for the planning and execution of these tests.

#### 2.4 INDEPENDENT VERIFICATION AND VALIDATION

While independent verification and validation (IV&V) is usually handled as a single activity, the two components, verification and validation, serve different objectives and involve different techniques. Verification is defined as:

The process of determining whether or not the products of a given phase of the software development cycle fulfill the requirements established during the preceding phase;

while validation is defined as:

The process of evaluating software at the end of the development process to ensure compliance with software requirements.

These definitions are extracted from the IEEE Standard Glossary of Software Engineering Terminology [IEEE83] and are therefore couched in software terms but can be broadened to cover the total computer system by substituting that term where software appears in the above quotations. An overall computer dictionary which may include this extension is still in preparation (IEEE Standards Project 610).

It is seen that verification is applied at the detail level and is an activity that can be aided by requirements analyzers and similar software tools that establish logical correspondence between a requirement and its implementation. Verification has several phases, e. g., verifying a specification against requirements, the design against the specification, and the code against the design. Validation, on the other hand, is an effort of broad scope that is usually carried out by use of simulations or of use of the unit under test in an environment simulator. In general usage the objective of validation is to determine that all requirements for the intended end use of a system are being

met. As specifically applied to flight critical systems the objective is to determine that all aspects of reliability, fault tolerance, and safety are being met.

Because of the sensitive nature of the verification and validation of a flight critical system, it is customary to keep all contractual aspects of this effort separate from the development. If IV&V is not carried out by the government itself, the responsible organization may have to certify that it is financially and administratively independent from the developer. This independence is essential for sound management, but it may carry a stiff price tag because it means that two separate teams have to be educated in the requirements, specification, implementation and test of the system. In addition, there will be conflicts between the developer and the IV&V team which will have to be resolved by the contracting agency, placing an additional burden on them.

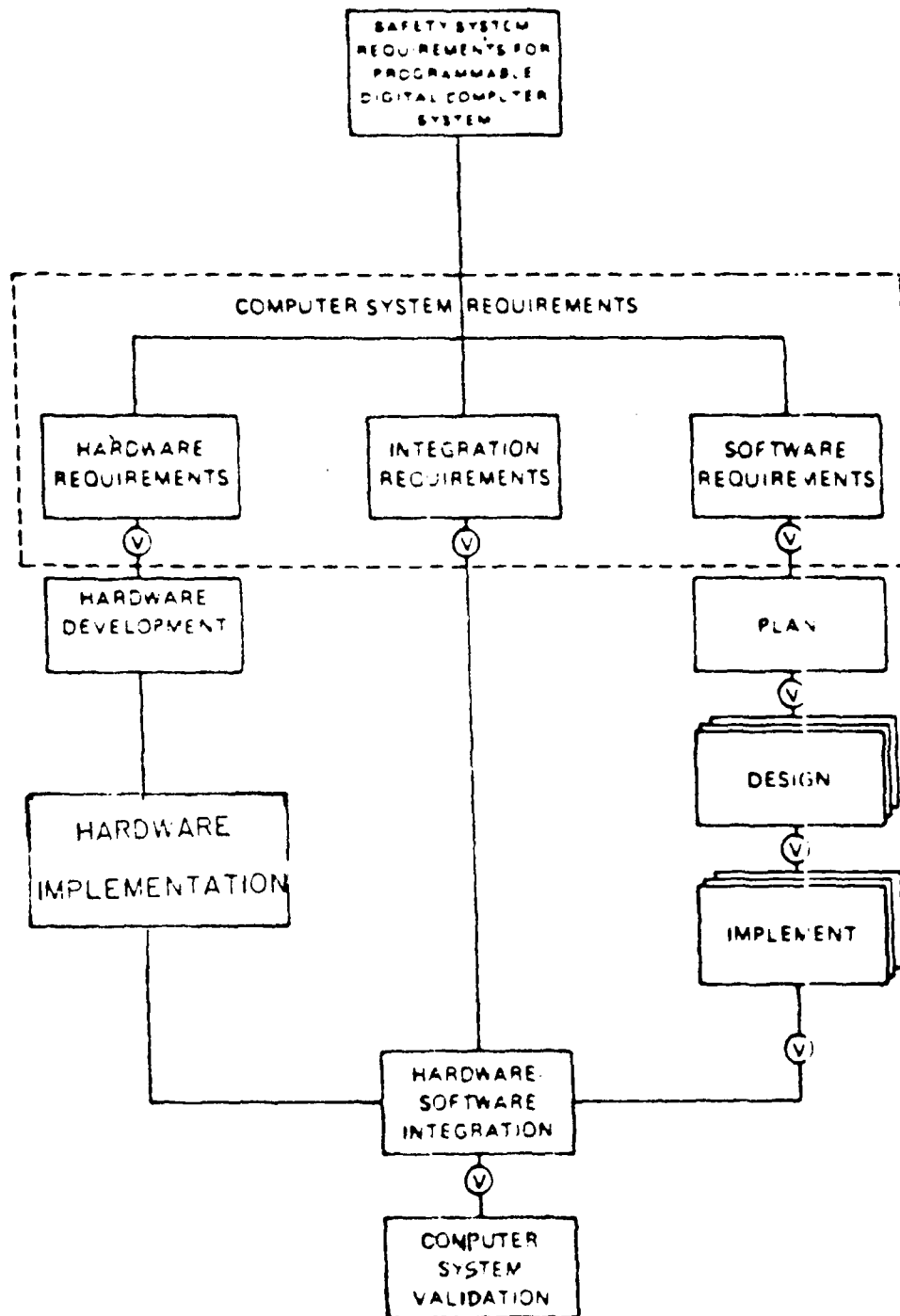
Considering only the contract costs, it has been reported that software IV&V cost runs between 12% and 69% of the development cost [RUBE75]. The lower limit applies for a very large project with completely defined requirements where the IV&V contractor personnel had prior experience on a very similar project. The high end of the range applies to small projects (less than 32k instructions) where requirements were not clearly defined and the IV&V personnel were not highly experienced. The size of avionics software typically falls close to the latter case. Where IV&V encompasses both hardware and software the ratio of IV&V to development cost may run lower than in purely software programs because of the large fraction of hardware costs associated with manufacturing.

Successful IV&V requires that the effort be started soon after development gets under way and well before the Preliminary Design Review. The IV&V contractor must be completely familiar with the requirements, the system, hardware and software specifications generated by the developer, and with the test planning prior to the PDR. IV&V tasks should include verification of the specifications against the requirements generated by the contracting agency. As the development progresses, the IV&V team will verify the design, the implementation, the pre-test documentation and the test of the system.

Because hardware and software components of the development may proceed largely independent of each other, the IV&V effort may have to adopt a similar structure but there must always be a core that focuses on the overall requirements and on the integration. A graphical representation of IV&V activities when hardware and software development is separated is shown in Figure 2-1. This structure has been applied to the validation of safety systems in nuclear power plants, and the figure is adopted from IEEE Std. 7.4.3.2 "Application Criteria for Programmable Digital Computer Systems in Safety Systems of Nuclear Power Generating Stations".

## 2.5 COMPLIANCE WITH RELIABILITY AND FAULT TOLERANCE REQUIREMENTS

The discussion in this section includes both the statement of compliance provisions and the enforcement of these. Requirements levied in a statement of work or in a specification are not ordinarily meant to be directly enforceable. As an example, there may be a requirement to avoid single points of failure. In the absence of separately stated compliance provisions it would take an actual



NOTE (V) REPRESENTS "VERIFICATION"

FIGURE 2-1 EXAMPLE OF TOTAL SYSTEM V&V

is there to show that the requirement has been verified. The need for separate compliance provisions is recognized in the Military Standard for Specification Practices, MIL-STD-490, e. g., Section 4.4 in the body of the standard and in sections xx.4 of the appendices (where xx denotes the first digital group of each appendix). Appendices I through III may be applied to computer systems or major components thereof. Appendix VI relates to software products, and appendices VII through X may be used for individual hardware items.

In each specification at least the following should be stated to ensure compliance:

- One or more specific methods for determining compliance;
- Responsibility for implementation of the method (usually this is assigned to the development contractor but it may also be assigned to the IV&V contractor);
- Location of activity;
- Time (either a calendar date or referenced to a review); and
- Reporting of results.

Suitable methods for establishing compliance in connection with computer systems include:

- Analysis -- usually documented in a technical report and discussed at a program review;
- Audit -- a determination that specified characteristics are present, usually documented in terms of a signed checklist;
- Review -- a determination that the item meets requirements, usually of broader scope than an audit, and documented in a report or minutes of the review;
- Simulation -- the article to be examined is connected to a simulated environment which can range from an all software simulation to a complete flight simulator. The results are usually documented in an engineering report. It is important to require that the veracity of the simulation itself be demonstrated and documented;
- Functional test -- test of the item against its functional requirements in accordance with an approved test specification and usually documented in a test report;
- Quality and stress tests -- testing to determine the reliability, design margins, endurance, and ability to tolerate unusual environments, usually documented in test reports and a summary engineering report; and
- In-service warranties -- Warranty that the item will perform under specified service conditions and for a specified time in accordance with the requirements.

In the evaluation of these methods it must be kept in mind that those applicable to the earlier development phases (notably the first three) are usually the most

valuable ones because they point to problem areas at a time when corrective steps can be taken without major impact on development cost and schedule. Simulation, test, and in-service warranties may appear to be more conclusive than the earlier methods but usually only prove that something has gone wrong that is very difficult to correct. The existence of extensive test requirements and of in-service warranties should in no way detract from insistence of a full complement of analyses, audits, and reviews. Both the contractor and the Government benefit from early detection of difficulties in complying with a requirement for flight critical systems.

In spite of the desire to hold the developer responsible for non-compliance with a specification, it is frequently impossible to resolve the issue on a purely legal basis. Contractors justifiably limit their responsibility to correction of deficiencies whereas the major impact on the developing agency arises from consequences of the deficiency, e. g., delay in schedule, inability to verify an interface, etc. Also, the requirements imposed on the contractor may be incomplete or inconsistent so that the responsibility for remedial action cannot be clearly established.

For these reasons some flexibility is usually necessary in the enforcement of the compliance provisions. In particular, the Government must be prepared to undertake additional analyses during the development, either by the developer or by another organization, when difficulties arise in connection with a flight critical item. A small budget reserve for such contingencies can avoid major delays when a difficulty is encountered that cannot be resolved within the existing contract framework.

## Chapter 3

### FAILURES IN DIGITAL EQUIPMENT

In the spirit of "Know thine enemy" this chapter provides information on the nature of failures in digital systems and how this can affect flight critical functions. The chapter starts with a classification of failure types which calls attention to the fact that random hardware failures, the subject of most of the conventional reliability literature, are only a small component of the failures encountered in most digital systems.

The second section discusses current experience with electronic control systems in aircraft. Very few digital systems are currently in service, and most of these serve functions which are either non-critical or are of limited authority. No catastrophic aircraft failures due to on-board digital equipment have been recorded in USAF and FAA documents that were made available for the preparation of this handbook. However, as pointed out in the third section, drastic changes are occurring in the use of full authority (and therefore critical) systems, and advances in digital device technology are further shrinking the dimensions of the basic building blocks of the digital systems. The latter step may introduce new failure mechanisms, and the wider usage brings with it the potential for heightened consequences of a failure. Continued concern and extreme care in the application of computers to flight critical functions is therefore indicated.

The final part of this chapter is devoted to failure and reliability models and introduces a uniform model for hardware and software failures. The model emphasizes the dependence of the occurrence of failures on the presence of faults (in hardware and software) and on the arrival of events in the environment that trigger these faults into producing failures. The model is also useful for describing the failure process at several levels of abstraction, starting with the physical and progressing through logic and information levels to the external level (i. e., manifestations in the aircraft) at which failures are ultimately observed.

#### 3.1 CLASSIFICATION OF FAILURES

As pointed out at the beginning of the preceding chapter, designating a component or system as flight critical carries with it an obligation to eliminate the causes of failure or to circumvent the effects of the failure at a higher system level. The classification of failures discussed here supports both of these approaches. Knowledge of where failures arise, how long they persist, and what their effects are is obviously important in guiding efforts at reliability improvement. Less obviously, perhaps, this information is also essential to devise appropriate fault tolerance provisions.

The classifications presented here include:

- Area affected;
- Cause of failure;
- Persistence of failure;
- Severity of effects;
- Multiple failure events; and
- Environment in which the failure was observed.

The discussion of these classifications is intended to emphasize the many-faceted nature of failures in computer based systems, to discourage dependence on a single failure rate as an indicator of system reliability, and to encourage the use of these classifications in the failure reporting for flight critical systems.

Many schemes for reporting failures in digital systems start with a division based on the area affected, hardware or software. A third category is desirable to record failures in the fault tolerance mechanisms which usually involve both hardware and software. Failures due to performance deficiencies, personnel actions, or interface with other systems should be considered in an "other" category.

In at least one current Air Force application different failure report forms are used for hardware and software failures. This procedure is motivated by the difference in corrective actions required but it is not the most desirable approach from a technical point of view. At the time of the incident there is not always complete certainty as to the cause, and even if the immediate cause is known (e. g., failure of a software routine to terminate) this might still be associated with a failure in the other category (e. g., a transient memory failure). A unified failure reporting system makes it easier to detect correlated hardware/software failures, and, as will be shown later, prevention of these is an important concern for flight critical systems.

A further classification of causes will distinguish between random, design, wearout and induced failures. The random failure concept is applicable to hardware items only and implies that the true cause of the failure is a design, wearout, process, or application deficiency which cannot be prevented or even clearly identified with currently available techniques. The random nature of the failure process makes hardware redundancy the choice method of circumventing this type of failure, and, conversely, this is the primary cause of failure for which redundancy, using identical part types, is effective.

Design faults, as the designation implies, will be found on all implementations of a given design. The term is used here in a broad sense and includes deficiencies in the requirements, specifications, process definition, and test and installation procedures as well as in the design proper. Because of the complexity of digital systems it is impossible to run exhaustive tests that would preclude the acceptance of articles that exhibit design faults. Also, design faults can result in hardware items that are weak rather than completely deficient in a given characteristic. Thus, many cycles of use may be required

before a failure can be observed. Design faults can be the cause of hardware, software, and other failures, and they are considered to be the sole cause of software failures. Because all units of a given design are affected, redundancy using identical part types is not effective against this failure mechanism, at least in principle. The latter qualification is inserted because where the failure is due to weak components (as identified earlier in this paragraph) or to interactions with the environment (see 3.4) redundant components of the same type may provide some degree of fault tolerance because they won't all fail at the same time. Wearout failures can be considered as due to special type of weakness inherent in the design which limits the useful life of the component. They are primarily found in mechanical or electromechanical parts. Electromigration, a process in which particles of a conductor are carried away by the current flowing in it, is a potential wearout mechanism in microcircuits that is very rarely encountered in current devices but could become a concern as device dimensions decrease. The preferred method for dealing with wearout failures is to replace the affected parts prior to the onset of wearout. Redundancy can provide some protection against the effects of wearout failure, particularly if the components do not have identical service time.

Induced failures are due to external events that cause specified interface characteristics to be violated. Failures due to fire, lightning, sabotage, or battle damage fall into this category. Random or design failures in one component can induce failures in a connected unit, e. g. by causing excessive voltage to appear at a data interface or by causing reduction in the supply voltage serving both units. The prevention of the propagation of failures in this manner is mandatory for the protection of flight critical functions and is an important issue in integrated systems. A combined classification of causes of failures, based on the preceding discussion, is shown in Table 3-1.

TABLE 3-1 CLASSIFICATION OF CAUSES OF FAILURE

Cause Classif.	Area Classification			
	Hardware	Software	Fault Tol.	Other
Random	x		x	
Design	x	x	x	x
Wearout	x			
Induced	x			x

Failures can be further characterized by their duration as permanent or transient. Intermittent failures are a special type of transient failure that repeats within the observation interval. The reversion to normal operation after a temporary failure can be spontaneous (e. g., by the reversal of the failure mechanism), automatic (e. g., through exception handling in a software routine), or initiated by the crew (e. g., by pushing a restart



button). If program restart is provided for and utilized most software failures will be in the transient categories. Even a transient failure can produce a permanent effect at the system level if there is a significant recovery period, e. g., the control system might go unstable or hard over. Permanent failures usually have a more severe impact on safety of flight or mission success than temporary ones. On the other hand, permanent failures are easier to diagnose and repair than transient ones.

The classification of failure effects can be based on computer manifestations or on mission effects. The latter are closely related to the criticality criteria and are discussed in Section 4. The computer manifestations can be characterized as bad data in a single task, abnormal termination of a single task, bad data for multiple tasks and abnormal termination of multiple programs. In terms of computation for flight critical functions, the latter two effects are much more undesirable than the former ones because they imply a propagation of failure effects beyond a single task. Substantial efforts are therefore warranted to prevent these effects.

Many fault tolerance mechanisms are designed to deal with a single manifestation of failure at a time. A common example is the single bit error correcting code used to protect against memory failures. These provisions are unable to cope with multiple malfunctions. The investigation and prevention of multiple failures is therefore an important task in the management of flight critical systems. Multiple failures may occur due to chance coincidence of two events that do not have a common cause. These are called uncorrelated multiple failures. In a system that has a low intrinsic failure rate the occurrence of uncorrelated multiple failures is extremely improbable. By far the greater portion of multiple failures are either induced (as discussed in a previous paragraph in this heading) or they are correlated. The latter implies that there is a single initiator (which is usually not observable) for the simultaneous events. A common example is the failure of a clock line that serves multiple functions. Obviously, fault tolerant design must guard against causes of correlated failures, and techniques for accomplishing this are described in Section 5 of this Handbook. However, one aspect of this is so important that it warrants repetition: the avoidance of any common elements between a monitor and the device being monitored.

The failure process is modeled in this Handbook as resulting from the joint presence of a fault (in hardware or software) and a triggering event (arising in the environment). Further details on this representation are described in 3.4. Because this model emphasizes the role of the environment in causing failures, a classification of the environment is necessary to fully characterize the failure process. Three major categories are found useful for this characterization: test environment, routine operational environment (further classified by the level of the prevailing workload), and operational environment while in an exception handling state. Consideration of these categories will help in arriving at a realistic assessment of the probability of failures, particularly with regard to hardware/software interactions during recovery from failures in a fault tolerant system.

### 3.2 EXPERIENCE ON CURRENT SYSTEMS

Automatic flight control systems in general and computers in particular have not been significant causes of aircraft accidents in the civil aviation field. A summary of 339 accidents in the 1959 - 1968 period contains no mention of automatic flight and engine controls as causes. Approximately 2/3 of these accidents were due to aircrew errors, while sabotage and airframe (including power plant) failures were tied for second place, each accounting for approximately 1/8 of all accidents [CLIF70].

A summary of 25 accidents and incidents in which flight controls (in general, not restricted to automatic systems) were implicated was obtained for this study from the National Transportation Safety Board. These events took place between 1972 and 1981. Automatic controls do appear as a cause in these but are a minor contributor compared to hydraulic and mechanical failures as far as frequency of failure is concerned, and compared to pilot error as far as severity of failure is concerned. A summary of the data is shown in Table 3-2.

TABLE 3 - 2

#### CIVIL AIRCRAFT ACCIDENTS/INCIDENTS INVOLVING FLIGHT CONTROLS (1972-81)

Primary Cause	Events		Fatalities	Injuries	Fatal. Inj.	
	No.	%			No.	%
Autom. Contr.	3	12	0	2	2	2
Mechanical	8	32	1	4	5	4
Hydraulic	7	28	0	0	0	0
Electrical	4	16	0	2	2	2
Pilot	2	8	88	17	105	90
Instruments	1	4	0	2	2	2
Total	25		89	27	116	100

The three events in which automatic flight control systems were the primary cause involved one pitch control computer failure (in a DC-10, presumed to be an analog computer), an unintended engagement of a yaw damper due to a wiring mistake, and a yaw damper failure due to a coffee spill in the cockpit which caused a short circuit in a connector. The sample is too small to draw any statistical conclusions, but it is fairly typical that two out of the three failures in the automatic control systems were due to external causes, one an induced failure (spilled coffee), and one a mistake during installation (which would be classified in the "other" area by the scheme described in the preceding section). This indicates the importance of protecting critical

systems against such incidents.

In the evaluation of data on current flight control systems it must be kept in mind that these are not flight critical except under unusual conditions (e. g., Category 2 or 3 instrument landings). Thus, failures in the digital components can be overcome by simply disengaging the affected control function and completing the flight in a back-up mode. These failures are corrected by routine maintenance action and data on their frequency are not available in the public domain.

Some pertinent further data can be gleaned from reports of the flight testing in the AFTI/F-16 program. This project is aimed at increasing the performance and maneuvering capabilities of the aircraft. Extensive use is made of digital flight control functions to achieve these goals. Reports are available on the first 118 flights conducted in this program which involve approximately 200 flight hours [MACK83a, MACK83b]. During this period there were several discrepancies in the performance of the automatic flight controls such as must be expected in an experimental program. These could be corrected by adjustment of parameters in the operational flight program. A number of 'nuisance' malfunctions occurred due to synchronization problems in the three parallel channels of the digital flight control system. These were eventually corrected by software changes. The most serious failure encountered involved the disengagement in flight of two of the three DFCS channels due to a software problem. The fault may have been introduced in the process of making a change and it was not detected by an otherwise rigorous test and configuration management system. Although the reports mention a number of hardware problems these were all related to the design (primarily time skew between the three redundant computers). No random hardware failures seem to have been encountered in flight.

The experience on the AFTI program also provides insight into the capabilities and problems with built-in test (BIT) functions. BIT detected two failures, one in an actuator and one in memory chips but there were numerous 'nuisance' indications from BIT. These were attributed to electromagnetic interference (EMI) but the possibility that they represented a correct response to transient or intermittent failures is not precluded. Further data on the use of flight critical digital systems and reliability experience with these is presented in Appendix D.

### 3.3 EXTRAPOLATION TO FUTURE SYSTEMS

The scarcity of current data on catastrophic aircraft accidents due to failures in computer based systems cannot be taken as assurance that such failures will not become a frequent occurrence in the future. There are comparatively few digital flight or engine control systems in operational use today, and these are (with the exceptions noted above) not flight critical. In contrast with the prevailing conditions, it is expected that first the military services and later civil aviation will in the future use aircraft which are heavily dependent on automatic controls for stability, maneuvering, and optimum utilization of the power plant. The urgency of achieving and demonstrating reliability that will support fly-by-wire techniques has prompted specialized studies and conferences on this subject [HOPK78, WENS78, RANG79, NASA79, LARI81].

Significantly increased dependence on digital control is also seen in the engine control field. In order to achieve fuel efficiency and very high power output it is necessary to modify the inlet geometry, the guide vanes, and the aft configuration. The actuation of these controls must be coordinated with turbine speed and fuel flow, and only a computer based system can perform adequately in this environment. Failure of the controller can lead to destruction of the engine or significantly reduced power output, both of which will jeopardize the safety of the aircraft [BAKE82].

In addition to the qualitative and quantitative increase in requirements for ultra-reliable digital control systems there are trends at work in electronic circuit design and fabrication which represent an equally startling departure from present practice: VLSI and VHSIC semiconductor devices. These can provide greatly increased density and operating speed for digital circuits and will undoubtedly be the building blocks for future avionics systems. The problem in the utilization of these devices is that very little is known about their failure modes. There is particular concern that to take advantage of the high device density, multiple functions will be placed on one chip, and that these functions will then be subject to correlated failures. A number of approaches are possible for reducing the incidence of common failures but none have been evaluated under operational conditions [HECH82].

Another area of concern is the software and firmware (programs that are stored in a permanent memory). Software is seldom cited as a cause for in-flight failures in digital flight control systems. Possible reasons for this include:

- Software for flight critical applications is much less complex than that for air traffic control, electronic (telephone) switching systems, or electronic funds transfer. Flight programs are typically several thousand bytes long while the programs utilized in command and control systems are several million bytes long.
- Because of its criticality the software receives extensive review and test, and because of its lack of complexity a given amount of effort can achieve a very thorough verification. Also, flight software tends to be stable (is rarely changed) whereas periodic and unscheduled updates are a fact of life for most other applications. A high failure frequency is usually encountered after an update.
- The input data for flight software contain fewer exception conditions than those for other applications. This is particularly true for commercial aircraft which fly the same routes under approximately the same atmospheric conditions day after day. Even military aircraft follow a specified flight plan in most cases. Within a given flight program the sequence of operations is usually fixed (under control of a scheduling algorithm). This is not necessarily so in other programs where a complex interrupt structure may be in use.
- In-flight failures may go unobserved unless they create highly unusual conditions. Software failures in attitude control or engine control systems may manifest themselves as slight deviations that cannot be readily distinguished from atmospheric disturbances or fuel flow problems. Even failures that cause an automatic system to disengage are not always recognized as such. They may be regarded as nuisance cut-offs or, where several crew members are involved, thought of as due to an improper action

by other personnel.

- It is frequently difficult to diagnose a failure as being due to software because of inability to duplicate the conditions that caused it. Many unidentified failures in avionic systems are suspected of being due to software.

In a study of software reliability for digital flight controls conducted by SoHaR for the NASA Ames Research Center, a comparison was made of the fault density (percentage of statements found to contain faults) in flight control programs and in a number of other (mostly non-aircraft) programs [HECH83B]. As shown in Table 3-3 the fault density in flight control programs was greater than that in the total program population. All of the programs in this sample were developed during the same time period (1977 - 1981) and the fault density was evaluated at approximately the same stage of program maturity. This finding makes it likely that difficulties of observation and diagnosis are significant causes for the low incidence of reported software failures in digital flight systems. The primary purpose of the table in the original reference was to show the benefits of the use of a high order language (HOL) on the quality of computer programs, a finding that is also of interest in the Handbook. Further evidence of the importance of software to flight critical systems is seen in the AFTI/F-16 experience that was reported in the preceding heading.

TABLE 3 - 3 EFFECT OF LANGUAGE ON FAULT DENSITY

Program Attribute	Assembly	HOL
No. of programs	6	15
Program size*	100k	1,124k
Fault density		
All programs	1.03%	0.15%
Flight controls	1.58%	0.52%
Range of f. d.	0.15% - 5.21%	0.01% - 0.86%

\* Equivalent executable assembly statements

### 3.4 A UNIFIED MODEL FOR FAILURES IN DIGITAL SYSTEMS

Most reliability models have been based on a block diagram representation of which figure 3-1 is an example. For elements in series the reliability of a combination of elements,  $R_s$ , is given by

$$R_s = \prod_{i=1}^n R_i \quad (3-1)$$

and for elements in parallel it is given by

$$R_p = 1 - \prod_{i=1}^n (1 - R_i) \quad (3-2)$$

The reliability,  $R_i$ , of the individual elements is derived from past experience or is predicted by methods identified in Military Standard for Reliability Modeling and Prediction, MIL-STD-756, with parameters for the prediction obtained from the Military Handbook for Reliability Prediction of Electronic Equipment, MIL-HDBK-217.

Independent of that approach there had been models for predicting the probability of failure of individual elements subject to a specific stress or load which take the general form shown in Figure 3-2. This methodology had originated in the structural field [FREU45] and had been adopted for electronic components about ten years later [LUSS57]. It has also been applied to mechanical and electromechanical equipment [KECE64]. The probability of failure is obtained as the convolution integral of the load and strength distributions, and the procedures for this are described in the references. For the present purpose it is sufficient to know that the failure probability is a direct function of the overlap of the two curves (the shaded area in the figure). This probability can be reduced by increasing the distance between the means as well as by reducing the width of the individual curves (usually accomplished by reducing the standard deviation of strength). This model provides insight into the failure process and indicates that the probability of failure can be reduced by lowering the load as well as by increasing the strength of the item.

There had been no formal attempt at combining the two models to reflect the interaction of load and strength at the system level although some aspects of the environment are taken into account in the reliability prediction parameters in MIL-HDBK-217. In the reliability assessment of fault tolerant computers it is important to model environmental effects explicitly because frequently the same stresses affect multiple elements or a monitored element and the monitor. Also, there is a need to identify (and protect against) failures at several levels of abstraction. During the IEEE 1980 Workshop on Validation of Fault Tolerant System the need for expressing these interactions was recognized and the failure model shown in Figure 3-3 was generated. The principal contributors were W. C. Carter, H. Hecht, and A. L. Hopkins.

According to this model a failure arises when a fault (hardware or software) interacts with an event in the environment which will be called a trigger. The effect of the failure is an error which may or may not be detected. This model shows that the number of detected errors can be affected by the inherent fault content of the equipment, by the imposed environment, and by the thoroughness of the observation.

An application of this model to several representation levels (see below, 1) of a pitch axis failure in a flight control system is shown in Figure 3-4. The

---

1. The designation of these levels is based on [AVI782]

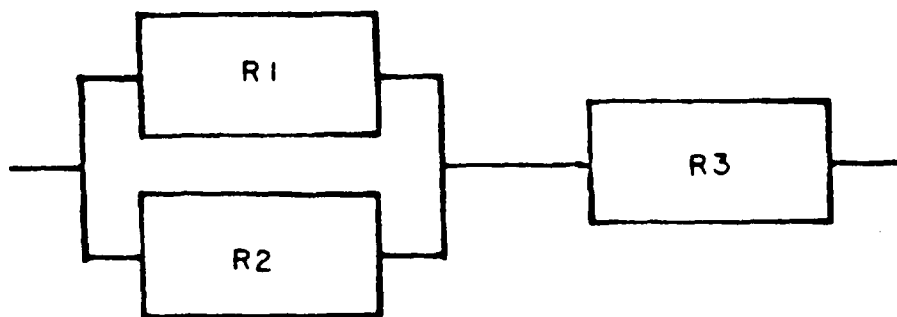


FIGURE 3-1 CONVENTIONAL RELIABILITY DIAGRAM

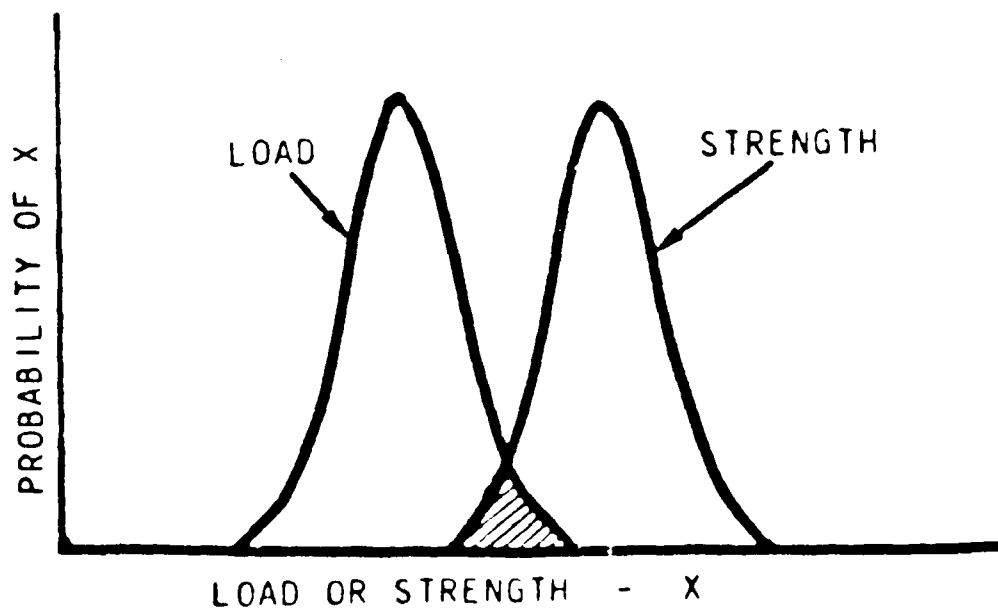


FIGURE 3-2 LOAD-STRENGTH MODEL

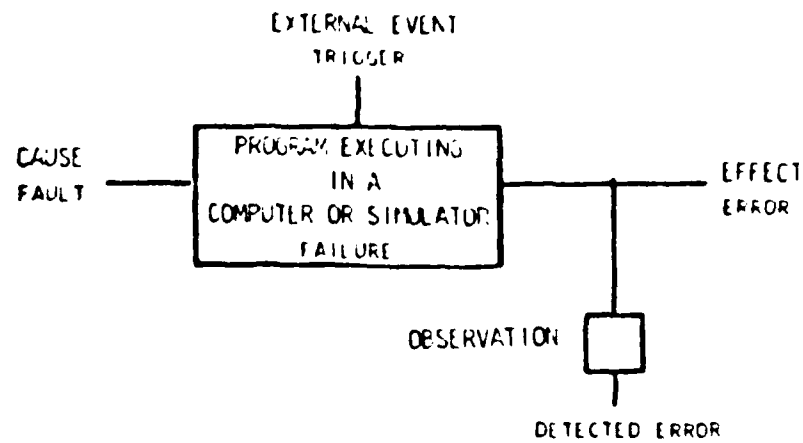


FIGURE 3-3 BASIC FAILURE MODEL

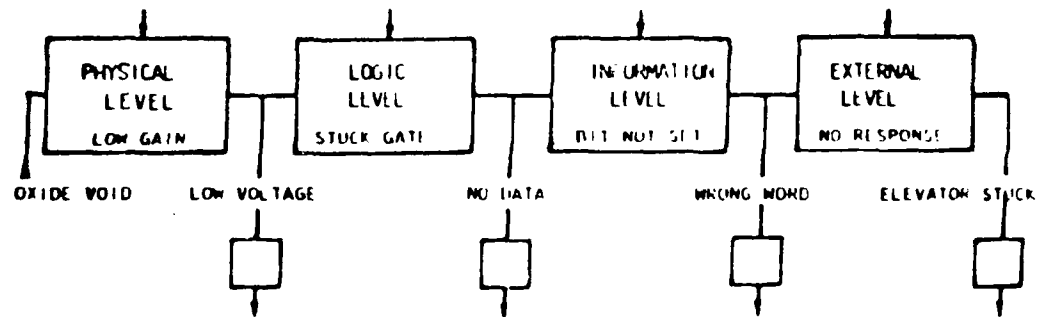


FIGURE 3-4 MODELING A PITCH AXIS FAILURE



initial fault is a void in the oxide layer of an integrated circuit which causes one particular transistor to exhibit low gain. The immediate effect of this failure at the physical level is a low output voltage, and this is usually not observable unless the semiconductor device is being subjected to special tests. The low gain appears as the fault to the next level, where the voltage is insufficient to initiate the transition of a gate. It is here assumed that the output of the gate was intended to set a bit in a register which in turn would transmit the new value of the elevator deflection command. At the logic level the effect of the failure is that required data are not available. This can be observed only if special test instrumentation is used. The next level, the information level, represents the output of the computer. Because the enabling bit had not been set, the old value of the elevator command continues to be output which represents wrong information. At the system or external level the effect appears as a stuck elevator, an effect that is clearly noticeable to the pilot. A passenger might become aware of the malfunction in terms of the pitch instability in the phugoid mode which can be thought of as a still higher level of representation. The oxide void, low gate voltage, lack of data, and the wrong elevator command are all causes of the failure, although at different levels. Specification of the correct level for fault identification and fault tolerance is quite important in the management of critical systems and will be discussed in Sections 5 and 6.

At the physical level it is frequently very difficult to identify a trigger mechanism. In this example the oxide void might have been there since the device was manufactured, or it might have been created (or considerably enlarged) recently due to thermal shock or chemical processes. Typically, the gain of the device had been at the low end of the response level of the next gate for a long time. The trigger of the failure process can only be described at the logic level, e. g., simultaneous transition to low state in gates  $i-1$ ,  $i$ , and  $i+1$  (where the affected gate is designated as  $i$ ). At the information level the trigger might be identified as simultaneous altitude and heading change commands (which resulted in multiple gates changing state at exactly the same time), and at the external level it might be related to a change in navigation mode that in turn generated the combined commands. Thus, the triggers as well as the causes of faults may have multiple representations.

When fail-safe or fail-operative requirements are levied in a system specification it is important to identify the representation levels at which these are to be effective. A fail-operative capability at the physical or logic level does not necessarily result in a fail-operative capability at the higher levels because it does not include software and system interfaces which are included there. On the other hand, it is possible to obtain fail-operative attributes at a high level by utilizing multiple lower level items each of which has fail-safe characteristics. It is extremely difficult to design a computer to be fail-safe, and where this is a requirement at the external level an additional monitoring and disconnect component may have to be provided.

Figure 3-5 shows how complete fault tolerant systems can be represented by this model. In a fault tolerant system or component the occurrence of individual faults represents an event in the environment rather than a failure condition. It is therefore represented as the trigger rather than the cause

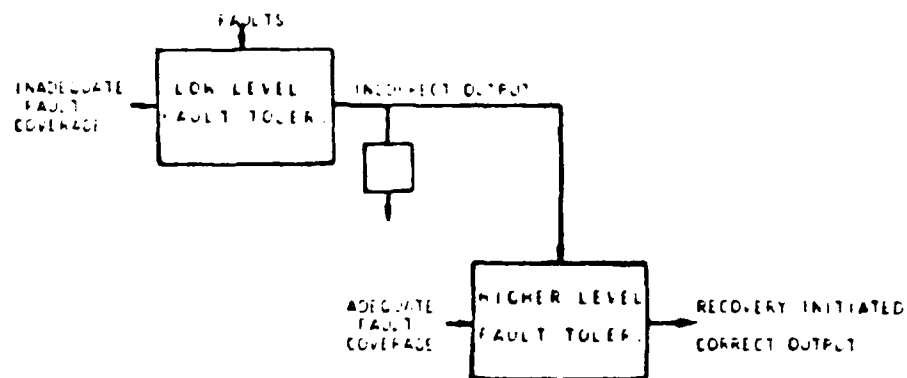


FIGURE 3-5 MODEL APPLIED TO FAULT TOLERANT SYSTEM

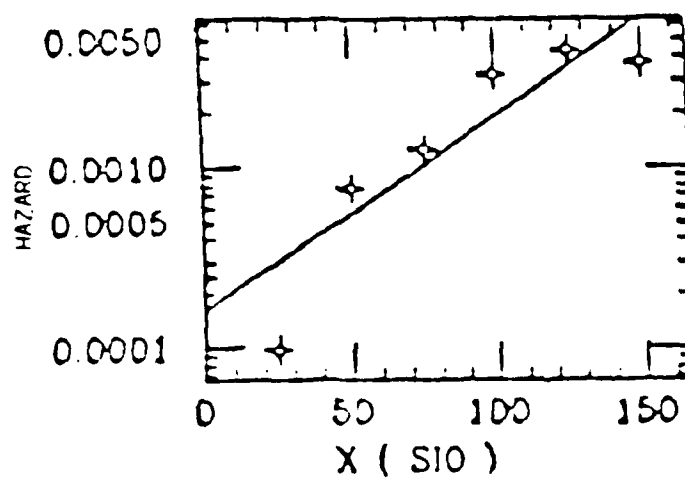


FIGURE 3-6 EFFECT OF INPUT/OUTPUT OPERATIONS

of failure of the system or component. Only where fault coverage is inadequate, i. e., where the design of fault tolerance was itself faulty, will there be an externally observable effect if a fault occurs. The figure shows a multi-level fault tolerant structure, a very desirable implementation that is further discussed in Section 5. In this example the consequences of inadequate fault coverage at a low level are masked by the correct performance of the fault tolerance provisions at the higher level, and an external observer will not detect any failure.

The basic model shown in Figure 3-3 is of value in establishing requirements for flight critical systems because it recognizes the dependence of failure probability on both the presence of faults in the equipment and the occurrence of events in the environment. When a heavy workload is imposed on a digital component the number of exception states that are encountered can increase much more than in proportion to the throughput. Exception states are conditions in which the computer is forced to delay the completion of a task or to use a less desirable capability. An example is a busy communication channel that requires that a task be suspended until needed data can be obtained. Exception states also arise from faults that are handled by existing fault tolerance provisions, e. g., error correction for memory or repetition of a garbled message. Most exception states arise in connection with input/output operations, and it is therefore significant that a recent study has shown a startling dependence of the failure hazard on the frequency of input/output operations [ROSS82]. Figure 3-6 is reproduced from the cited report. The hazard is defined as the probability of failure at the level of I/O activity at the abscissa value, given that no failure had occurred at a lower level of I/O activity. At the very least this dependence of failure rate on workload suggests that a substantial amount of testing be conducted at high activity levels. It also makes it desirable to provide sizeable performance margins in computers associated with flight critical functions so that fewer exception conditions will be encountered.

## Chapter 4

### FLIGHT CRITICAL SYSTEMS AND COMPONENTS

This chapter discusses the definition of critical systems, where they are likely to be found, and how criticality affects interfacing and integration of subsystems. In the first section a number of definitions of critical aircraft systems are presented and the terminology for this Handbook is selected. Next, the potential for encountering critical components in various Air Force aircraft types is investigated. A breakdown by mission phase (take off, climb, etc.) and equipment type (yaw damper, terrain avoidance, etc.) is used. In the third section the characteristics of critical systems are discussed in terms of how they interact with the primary aircraft controls. Classifications of direct access (to primary controls), indirect access, human mediated, and non-control functions are used in this connection. The fourth and fifth sections cover system interfaces and interfaces within a system (between digital components and sensors and actuators), respectively. The final section in this chapter investigates the effect of integration on critical systems, a topic that is becoming increasingly important in Air Force applications.

#### 4.1 CRITERIA FOR CRITICALITY

For aircraft that fall under the jurisdiction of the Federal Aviation Administration, a critical function is defined as one "... whose failure would contribute to or cause a failure condition which would prevent the safe flight and landing of the airplane (see below, 1)." In the same context, a failure condition is defined as

A consequential airplane state which has an impact on the functional capability of the airplane or the ability of the crew to cope with adverse operating conditions, or which would prevent continued safe flight or landing... A defined failure condition provides the criteria for classifying system functions as non-essential, essential, or critical ...

---

1. FAA Advisory Circular 25-1309-1, 7 Sep 1982

For military flight control systems a different terminology has been adopted (see below, 2): essential designates the most critical functions, followed by flight phase essential and non-critical. A flight control function is defined as essential "...if loss of the function results in an unsafe condition or inability to maintain FCS Operational State III." The latter state is associated with the minimum safe operation and is defined as:

Operational State III is the state of degraded flight control system performance, safety, or reliability which permits safe termination of precision tracking or maneuvering tasks, and safe cruise, descent, and landing at the destination of original intent or alternate but where pilot workload is excessive or mission effectiveness is inadequate. Phases of the intended mission involving precision tracking or maneuvering cannot be completed satisfactorily. This state satisfies at least MIL-F-8785 or MIL-F-83300 Level 3 flying qualities requirements.

The term flight-critical will be used here to designate functions whose continued operation is required to assure safe continuation of flight with moderate maneuvers and safe landing. A good many functions will be found to be clearly non-critical by this definition, and others will be identified as critical in the next Section of this Chapter. However, the classification of some functions will depend on the application (mission) of the aircraft, the environment (tropics vs. arctic), and the capabilities of the crew. There is no "erring on the safe side" because placing functions that are less essential into a higher classification will (a) dilute the effort that can be allocated to the essential functions, and (b) increase the number of flight critical interfaces, thereby inherently degrading the reliability of the most critical functions as explained in Section 4.4. Flight-critical functions should be defined as early in the life of a weapon system as possible to permit focusing available resources on the truly essential elements. Table 4-1 is a minimum checklist of factors that need to be known to delineate flight-critical functions. Other factors may be added to this list for specific applications.

TABLE 4 - 1 CHECKLIST FOR SELECTION OF FLIGHT CRITICAL FUNCTION.

1. Aerodynamic stability of the aircraft
2. All-weather/restricted weather operation
3. Ability to land in rough terrain, water, snow
4. Expected fraction of severe weather operation (arctic locations, hurricane research, etc.)
5. Expected fraction of terrain-following and other low-altitude operation
6. Expected availability of buddy aircraft for back-up of navigation and flight management functions
7. Requirements for damage tolerance (particularly for damage that reduces aerodynamic stability)
8. Minimum pilot skill levels

The utilization of these factors for the identification of flight-critical functions is obvious in most cases. The ability to land in rough terrain, water, or snow makes more alternate landing sites available and correspondingly reduces the duration for which emergency electric and hydraulic power needs to be provided. Operation in severe weather conditions imposes a heavy workload on the flight crew even in the absence of any equipment malfunction and reduces the ability to cope with any abnormal operation. Low altitude operation increases the need for automatic flight and engine controls and thus makes these functions more critical.

Lack of aerodynamic stability in general or for some flight conditions is a key factor in changing the role of automatic flight control systems from a mission essential to a flight critical function. Requirements for high maneuverability and high speed in combat aircraft make it impossible for the designer to achieve flying qualities that permit manual control by the pilot. At the same time the increased functional and performance capabilities of current digital flight control systems make them well suited for the control of aerodynamically unstable aircraft. Automatic flight control systems can compensate for stability and control difficulties due to damage sustained by an aircraft, and the enhancement of these capabilities is the objective of a major current effort the details of which are outside the scope of this Handbook. Although flight-critical functions represent a safety hazard as defined in MIL-STD-882, the definitions and provisions of that standard have only very limited applicability to aircraft systems. The hazard severities are defined in MIL-STD-882 as:

- Category I - Catastrophic. May cause death or system loss.
- Category II - Critical. May cause severe injury, severe occupational illness, or major system damage.
- Category III - Marginal. May cause minor injury, minor occupational illness, or minor system damage.

- Category IV - Highlighted. Will not result in injury, or loss of life, illness, or system damage.

Because the severity classification does not provide for differentiation on the basis of the number of flight phases affected, all failures in flight-critical systems will be classified as Category I. The expected frequency of encountering the hazard is taken into account in the criticality classification of MIL-STD-882 but not in a manner that would be normally consistent with the MIL-F-9490D definitions.

#### 4.2 CRITICALITY BY AIRCRAFT TYPE AND MISSION PHASE

There is increasing dependence on digital computers for flight-critical functions in all aircraft types. Factors responsible for this tendency include:

- Reduction in flight crew size;
- Decreased aerodynamic stability (to improve maneuverability and increase fuel efficiency);
- Demands of advanced weapon delivery systems; and
- Requirements for damage tolerance which imply an ability to fly with reduced aerodynamic stability or maneuverability.

In part the dependence digital technology for flight-critical functions is also caused by the availability of equipment that is highly reliable and which can enhance the capabilities of the aircraft in a very cost-effective manner. While the utilization of equipment that falls into this area can be arbitrarily restricted, doing so might deprive our forces of a technological edge.

From the foregoing it is apparent that the criticality assignments can at best represent typical current practice. For most advanced aircraft types additional systems or equipment may fall into the flight-critical category.

The following tables list typical flight-critical systems for several aircraft types in Air Force inventory. The abbreviations used to designate mission phases are:

TO Take-off	WD Weapon Delivery
CL Climb	ED Emergency or Power Descent
CR Cruise	IA Instrument Approach
SS Supersonic Flight	IL Instrument Landing

The instrument approach phase also includes other low altitude flight phases. In order to keep the tables uniform, these column headings have been maintained for all aircraft types. Where a given phase is not applicable, lower case letters are used (e. g., supersonic flight in case of helicopters).

TABLE 4 - 2 TYPICAL FLIGHT-CRITICAL EQUIPMENT FOR TRAINERS

Equipment	Flight-Critical for Mission Phase							
	TO	CL	CR	SS	WD	ED	IA	IL
Yaw Damper		x	x	x	x	x		
Flight Control with Coupler							x	x
Thrust Management or Monitoring System							x	x
Air Data System		x	x	x	x	x	x	x

The table addresses the needs for an advanced instrument trainer. Primary trainers do not usually employ digital components for flight-critical functions.

TABLE 4 - 3 TYPICAL FLIGHT-CRITICAL EQUIPMENT FOR TRANSPORTS

Equipment	Flight-Critical for Mission Phase							
	TO	CL	CR	SS	WD	ED	IA	IL
Yaw Damper		x	x					
Flight Control with Coupler	x	x	x				x	x
Thrust Management or Monitoring System	x	x	x				x	x
Auto Reverse Thrust								x
Inertial Navigation	x	x	x					
Air Data System		x	x				x	x
Flight Director	x	x	x				x	x
Communication & IFFN	x	x	x				x	x

The selection of flight critical functions for the Transport is based on equipment provided on recently designed commercial transports and on the C-17.



TABLE 4 - 4 TYPICAL FLIGHT-CRITICAL EQUIPMENT FOR SURVEILLANCE AIRCRAFT

Equipment	Flight-Critical for Mission Phase							
	TO	CL	CP	SC	WT	ED	IA	IL
Yaw Damper		x	x	x	x			
Flight Control with Coupler	x	x	x	x	x		x	x
Thrust Management or Monitoring System	x	x	x	x	x		x	x
Automatic Terrain Following/Avoidance			x	x	x			
Automatic Threat Avoidance			x	x	x	x		
Inertial Navigation	x	x	x	x	x			
Air Data System		x	x	x	x	x	x	x
Head-Up Display	x	x	x	x	x		x	x
Communication & IFFN	x	x	x	x	x	x	x	x

The systems indicated for the Weapons Delivery phase are those essential for the surveillance mission. The criticality assignments for Reconnaissance aircraft are identical to those listed above.

TABLE 4 - 5 TYPICAL FLIGHT-CRITICAL EQUIPMENT FOR FIGHTER  
AIRCRAFT

Equipment	Flight-Critical for Mission Phase							
	TO	CL	CR	SS	WD	ED	IA	IL
Yaw Damper		x	x	x	x	x		
Flight Control with Coupler	x	x	x	x	x	x	x	x
Thrust Management or Monitoring System	x	x	x	x	x	x	x	x
Auxiliary Surface Controls			x	x	x	x		
Automatic Terrain Avoidance			x	x	x			
Automatic Threat Avoidance			x	x	x	x		
Air Data System		x	x	x	x	x	x	x
Head-Up Display	x	x	x	x	x		x	x
Communication & IFFN	x	x	x	x	x	x	x	x

The auxiliary surface controls refer to leading edge or canard surfaces, swinging tail, etc. The configuration selection and monitoring for these functions is included in the control equipment.

TABLE 4 - 6 TYPICAL FLIGHT-CRITICAL EQUIPMENT FOR BOMBERS

Equipment	Flight-Critical for Mission Phase							
	TO	CL	CR	SS	WD	ED	IA	IL
Yaw Damper		x	x	x	x	x		
Flight Control with Coupler	x	x	x	x	x	x	x	x
Thrust Management or Monitoring System	x	x	x	x	x	x	x	x
Automatic Thrust Reverse								x
Automatic Terrain Following/Avoidance			x	x	x			
Automatic Threat Avoidance			x	x	x	x		
Inertial Navigation	x	x	x	x	x			
Air Data System		x	x	x	x	x	x	x
Head-Up Display	x	x	x	x	x		x	x
Communication & IFFN	x	x	x	x	x	x	x	x

TABLE 4 - 7 TYPICAL FLIGHT-CRITICAL EQUIPMENT FOR HELICOPTERS

Equipment	Flight-Critical for Mission Phase							
	TO	CL	CR	ss	wd	ed	IA	IL
Three-Axis Stability Augmentation	x	x	x				x	x
Flight Control with Coupler	x	x	x				x	x
Automatic Collective/Throttle	x	x	x				x	x
Automatic Terrain Avoidance			x					
Flight Director/Head-Up Display	x	x	x				x	x
Communication & IFFN	x	x	x				x	x

The coupler to be furnished with the flight control function typically includes automatic hovering capability, either by means of navigation inputs or by a physical connection to the ground ("rope trick"). The latter is particularly pertinent for rescue helicopters.

TABLE 4 - 8 TYPICAL FLIGHT-CRITICAL EQUIPMENT FOR UNMANNED AIRCRAFT

Equipment	Flight-Critical for Mission Phase							
	TC	CL	CR	SS	WD	ED	IA	IL
Flight Control with Coupler	x	x	x	x	x	x	x	x
Thrust Management	x	x	x	x	x	x	x	x
Automatic Terrain Following/Avoidance			x	x	x			
Automatic Threat Avoidance			x	x	x	x		

The equipment indicated above is suitable for an advanced type of RPV in which economic considerations dictate the designation of functions as flight critical. Smaller RPVs may be considered expendable and thus have no flight-critical functions.

#### 4.3 CRITICALITY BY AIRCRAFT SYSTEM

In the assessment of criticality for individual systems it is useful to distinguish between classes of systems based on their effect on primary aircraft controls (control surfaces and thrust level). The following classification will be used here:

- DIRECT ACCESS - Systems which exercise direct control, typically requiring a frequency response above 1 Hz.
- INDIRECT ACCESS - System which couple through Direct Access systems, typically requiring a frequency response below 1 Hz.
- HUMAN MEDIATED - Systems which provide a primary output to the pilot or other personnel.
- NON-CONTROL FUNCTIONS - Systems which furnish an output that is not explicitly related to a control function, e. g., communications.

Criticality aspects for each of these classifications are discussed below.

##### 4.3.1 Direct Access Functions

By their nature, functions which have direct access to the primary aircraft controls need the most careful review in all aspects of design and test. Representative functions in this classification are stability augmentation (including yaw dampers), automatic pilot, and thrust control. Deliberate attenuation of the high frequency response, which is useful in reducing the effects of malfunctions in other classifications, is usually not possible

because the direct access functions need to exercise the aircraft close to the inherent capabilities of the control loops in order to provide the required performance of stability and attitude or speed control. Limiting the control authority of the stability augmentation system can be useful in reducing the impact of hard-over malfunctions in the output circuits. Mechanical restrictions on actuator travel are a very desirable form of implementing limited authority. This usually requires providing a separate actuator for stability augmentation, and adjusting the midpoint of its operating range as a function of flight conditions by a low response auxiliary actuator or by a separate trim system. Other acceptable means of limiting authority are flow restrictions in hydraulic actuators and limit switches in electric actuators. Redundancy of actuators or of actuator controls (valves) can provide both fault tolerance and limited authority for some failure modes.

The following aspects of direct access functions need to be considered:

- Enable/Disable - Where there are switches that enable or disable these functions, great care must be taken to avoid unintentional operation and to protect against the consequences of mechanical or electrical failure of these components. Redundancy of the electronic and electromechanical equipment can be defeated by a single failure in the enable/disable portion.
- Permanent Active Failures - These are failures which, in the absence of protective measures, drive the output components to a permanent hard-over position. Redundancy with output voting is a very effective means for coping with this type of malfunction. In some cases the occurrence of this malfunction can be detected electronically before the actuator responds fully, and automatic disengagement or output limiting can be applied. Where this approach is selected, testing under a wide range of rates of approaching the hard-over condition must be undertaken. Where control system redundancy is implemented through split surfaces or dual engines, effective means must be provided for disengaging the failed part of the control system and either centering the affected surfaces (or engine actuators) or bringing them under control of the surviving part of the control system. The crew should be alerted in case of any active failure because the ability to tolerate further faults has been impaired, and in some cases there may be a reduction in maneuvering capabilities or in the performance of the power plant.
- Intermittent Active Failures - Provisions for coping with permanent active failures are usually also well suited for handling intermittent failures. However, the fault masking characteristics of a voting configuration might suppress the evidence of this malfunction so that it may not be repaired until it recurs under conditions that defeat fault masking (e. g., associated with a second failure) and that can also render the redundancy ineffective. It is important that full circumstances of voting disagreements be recorded in a non-volatile register that can be accessed by maintenance personnel, and that effective management controls be instituted to prevent dispatch of aircraft with unresolved voting disagreements.
- Passive Failures - Passive failures render the control system unable to perform its intended task but do not place the output elements (aircraft or engine controls) into an extreme position. The colloquial expression for a passive failure, "the system goes dead", summarizes the effects

quite well. In the absence of a redundant channel, a passive failure will cause complete loss of the function, e. g., stability augmentation, and thereby places an excessive workload on the pilot. In some cases this will lead to immediate loss of the aircraft and in others it will lead to loss of the aircraft if there are other circumstances which place heavy demands on the pilot. In all cases there is almost certain loss of mission capability. Passive failures are well tolerated by systems that employ split control systems, whereas active failures are usually tolerated by these only if they can be converted to passive failures by disabling the failed channel or restricting its authority. On the other hand, passive failures are more difficult to detect. Heartbeat monitors and watchdog timers are commonly used for detection of passive failures in digital systems but these do not provide coverage of the output circuits of the computer or control equipment. Capturing of information on passive failures for manual or automated maintenance logs is made difficult by the detection problems, and this area needs attention in development and evaluation.

- Mode Change Failures - Most digital systems can operate in several modes (e. g., target acquisition, lock-on, and break-off) and unintentional transitions between these modes can cause flight critical failures. Consider an aircraft on automatic instrument landing when the autopilot switches inadvertently to a high speed cruise mode. The control deflections commanded in the latter mode are completely inadequate to maintain the aircraft on the desired flight path, and such a failure is therefore likely to result in a disaster. Mode change failures may occur due to an internal malfunction in the computer or controller, and in that case they will probably be tolerated if they affect only one of a number of redundant channels. However, mode changes are frequently externally commanded, either by the pilot or by an interfacing system (weapons control system, air data system), and a faulty command from these sources is likely to affect all redundant channels. Repeated inadvertent mode and gain changes were experienced in a recent Air Force flight test program, fortunately under flight conditions which made them non-critical. They were caused by an interfacing device, but the exact nature of the failure (even whether hardware or software) has not been established. Specific means of protecting against these failures is discussed in connection with critical interfaces in the next section.

As indicated, redundancy with voting protects against most of the failure modes for this group. The most frequently encountered form of this type of fault tolerance is triple modular redundancy (TMR), although quadruple redundancy is now being introduced in some critical aircraft systems. Detailed characteristics of these configurations are discussed in the next Chapter.

#### 4.3.2 Indirect Access Functions

Functions which control the aircraft or engine indirectly, typically through inputs to the autopilot and engine controller, are in this category. Examples are navigation systems, weapons control systems and air data system inputs to the flight and engine controls. Air data system outputs that drive instruments are covered in the following subsection. Several techniques are available within the Direct Access digital systems to limit the effect of failures in

connected equipment:

- input comparison or voting -- where several identical inputs are provided;
- smoothing -- filtering of high frequency components, particularly those arising from a sudden active malfunction;
- limiting -- restricting the magnitude and rate of signals accepted for further processing; and
- analytical redundancy -- computing the expected value of an input from a related physical quantity, e. g., approximating radar altimeter input by a barometric altitude input that has been adjusted for terrain altitude.

The value of these protective measures depends on the response margin between the input function and the airframe capability. This margin is typically large for slowly varying inputs, such as navigation and instrument approach. Weapon control systems, on the other hand, may need to utilize the full performance envelope of the aircraft and hence do not permit much filtering and limiting. Voting and analytical redundancy techniques also require some smoothing to suppress normal differences in output from individual devices, but these techniques do not require large response margins to be effective.

The principal advantage of applying fault containment in the Direct Access components is that Indirect Access equipment does not have to be modified for service of flight-critical functions. In addition, limiting and filtering parameters may have to be changed as a function of the controls mode, and this change is much more easily handled in Direct Access components which are the principal locale of the mode changes.

The criticality of various failure modes is discussed below:

1. Enable/Disable - The criticality of enable/disable provisions depends on the nature of the system and on the flight condition. The worst case is represented by unintentional disablement of the automatic landing function while the aircraft is close to the ground under instrument conditions. At the other end of the spectrum, disengagement or accidental engagement of a navigation coupling mode will not normally lead to an unsafe flight condition. As a minimum, the engage/disengage status of all coupled functions must be clearly visible to the pilot. For some functions additional analysis and protection will be required, similar to those for Direct Access functions.
2. Active Failures - The significant means of protection against active malfunctions in coupled equipment have been mentioned in the introductory paragraph: comparison, smoothing, limiting and analytical redundancy. Logging of malfunctions (or suspected malfunctions) for maintenance purposes should be insisted on. Disengagement of the coupled function may be acceptable as a means of dealing with active failures, except for automatic landing and approach.
3. Passive Failures - The immediate effect of a passive failure is equivalent to an unintentional disengagement of the function. As indicated above, this can be tolerated in many cases. The more difficult aspect of this failure type is the lack of an explicit indication to the pilot that the function is no longer available or active. Heartbeat

monitors, common mode output (equivalent to the flag indication of a cross pointer meter), and analytic redundancy are suitable means of failure detection.

4. Mode Change Failures - Mode change failures in coupled functions are less likely to be flight critical than those in direct access functions. The most difficult problem is again posed by automatic landing where redundancy at all levels of equipment is normally provided for protection.

Most forms of redundancy offer good protection against failures in Indirect Access functions. The redundancy management (selection of active functions, setting of parameters, and reconfiguration when necessary) can be handled by the Direct Access equipment. Clear indications of the status of coupled functions must be furnished to the pilot to avoid commitment of the aircraft to flight conditions which are outside of the current capabilities of the equipment.

#### 4.3.3 Human Mediated Functions

In these functions digital equipment furnishes displays to the pilot who acts on these by exercising the primary flight or engine controls or by changing the setting of the autopilot or engine controller. A typical example is a head-up display or helmet sight. There is implicit dependence on the pilot's judgement for detection of invalid output from the function. Equipment redundancy may be managed by the pilot (e. g., by selector switches). Individual failure modes are discussed below:

1. Engagement/Disengagement - Accidental engagement is normally noticed by the pilot and signals displayed will be disregarded. Accidental disengagement will also be noticed if proper visual cues are present only when the function is engaged.
2. Active Failures - Failures of near full-scale magnitude will usually be detected by the pilot and the signals will be disregarded. Failures of a lesser magnitude may not be recognized unless there are failure monitoring provisions in the equipment. This type of failure presents the most difficult problems for Human Mediated functions.
3. Passive Failures - As in the case of Indirect Access functions, passive failures pose a threat of unsafe conditions only if they go unnoticed. Heartbeat monitors, common mode output, and analytic redundancy can be used for detection. A significant change in the display can be used to call the pilot's attention to the failed condition.
4. Mode Change Failures - Where the mode is explicitly indicated in the display, or where the mode change is otherwise discernible by the pilot, there is little likelihood of a catastrophic event resulting from an unintended change. Where the pilot may be unaware of the change, protective equipment is required similar to that identified for the previous functions.

Human Mediated functions normally permit an adequate response margin relative to the airframe capabilities. Also, the judgement of the pilot can filter out the



effects of some malfunctions. Therefore massive redundancy, in particular TMR, is not usually required. Where there are several flight crew members, independent computation and display equipment is desirable. Status indications and monitoring for passive failures are important for avoiding flight-critical situations.

#### 4.3.4 Non-Control Functions

Digital equipment is increasingly being utilized for functions which have no direct relationship to aircraft controls, the most prominent ones being digital communications, identification of friend/f /neutral (IFFN), and target designations. The failure mode most likely to cause a flight-critical situation is a permanent outage of one or more of these functions. The protection against this is found in physical or analytical redundancy, usually selected under pilot control, e. g., communications equipment for several frequency bands, dual IFFN, and depending on communications when the target designator link fails.

### 4.4 CRITICALITY OF SYSTEM INTERFACES

Functions which are themselves non-critical can cause failures in critical digital systems through four distinct interfaces: utilities, dedicated links, buses and software. Problems specific to each of these are described below. An obvious precaution, common to all of these interfaces, is to restrict the number of equipments that are directly connected to flight-critical systems. Unfortunately, the pressure for "integration" forces a higher degree of functional and physical connectivity which poses serious problems of interface control. Issues arising from the integration of functions that are not flight critical with flight critical systems are discussed in the final section of this chapter.

#### 4.4.1 Utility Interfaces

Utilities required for digital equipment always include electric power, and sometimes thermal control (coolant) or timing signals from a central source. Failures in the utilities can propagate to flight critical systems due to:

- Failure of the prime equipment for each utility (electric generator, refrigerator, or clock);
- Failures in the distribution system (wiring, tubing, connectors); and
- Failures in other user equipment (short circuits, excessive heat generation).

Ideally, utility inputs to flight critical systems should be uninterruptible, i. e., they should maintain their specified characteristics as long as the aircraft is not totally disabled. In the electric power area the ideal condition is only remotely approached in standard installations, and with regard to the other utilities there are no guidelines at all. Figures 4-1 and 4-2 show ac and dc voltage limits, respectively, for "abnormal operation" from MIL-STD-704D. The term in quotes includes conditions in which one primary power supply has failed and an automatic transfer to another supply is being accomplished. During the possible outage period of 7 seconds it is required that connected equipment;

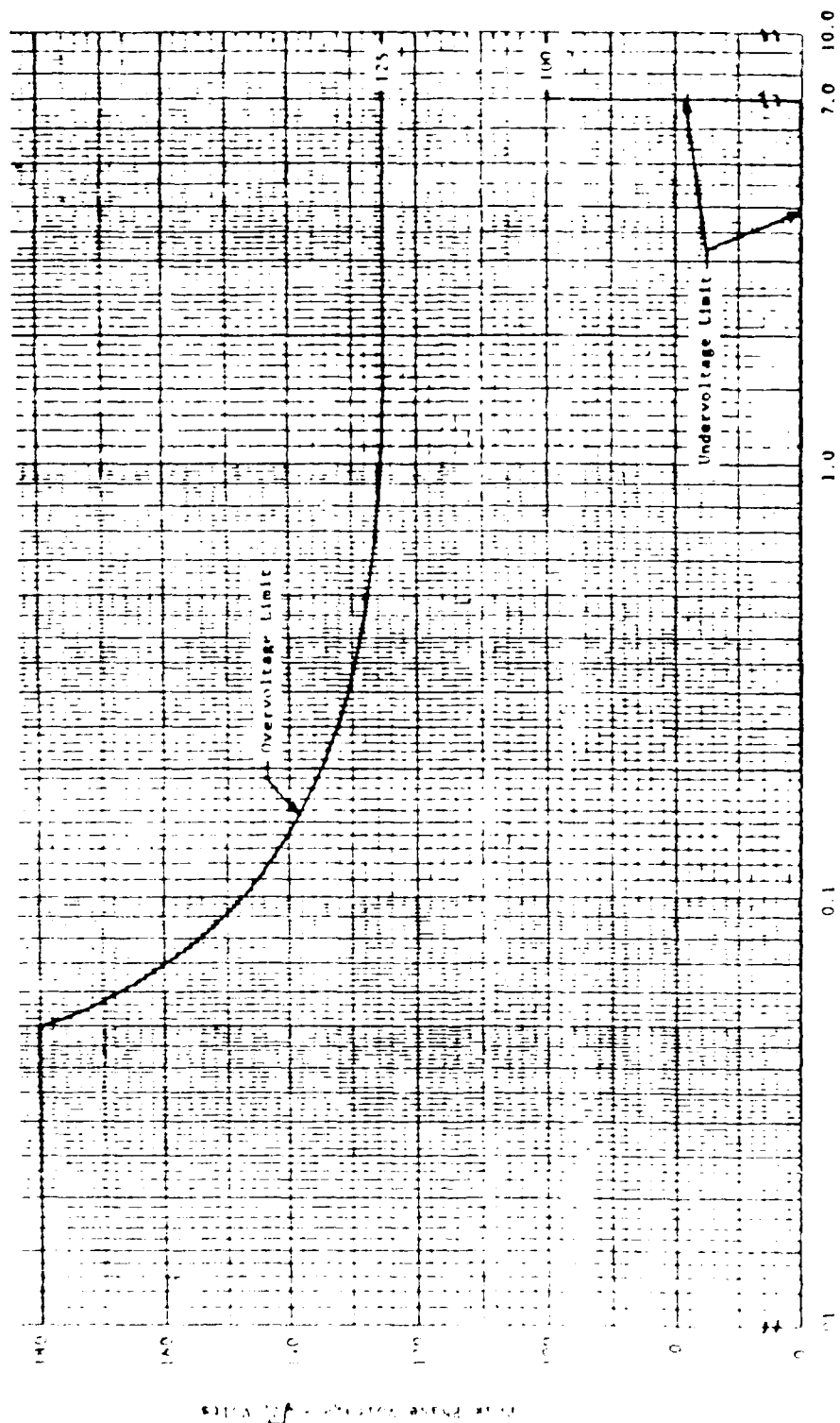
shall not produce a damaging or unsafe condition and shall automatically recover full specified performance when the electric power characteristics are restored to the normal operation limits herein.

Since any cessation of the operation of flight critical equipment can cause an unsafe condition, there is an implicit requirement that an alternative power supply be provided for the digital components to sustain operation for at least 7 seconds. Alternatively, redundant units of flight critical equipment may be connected to different buses. However, during periods of power transfer more than one bus may be in an abnormal state, and thus reliance on this approach needs careful analysis of the specific installation. Also, there is frequently only one bus that is kept free of switched high-power users and which therefore has a relatively low exposure to EMI. Representative implementations of electric power distribution in contemporary aircraft are described in Appendix E.

The power conversion unit in each flight critical digital component must be capable of tolerating one of the voltage envelopes shown in the figures (depending on whether the primary power input is ac or dc), and it must furnish a much more stable low voltage dc supply for the digital circuits. It must also suppress all high frequency components of the input voltage and present a very low source impedance to prevent propagation of internally generated high frequency components (due to the pulsating nature of the internal voltage drop). The high frequency components, if propagated, can be interpreted as digital signals by the circuits fed from the power conversion unit. The design of adequate power converters is difficult, and this aspect of computer engineering frequently does not receive the proper attention.

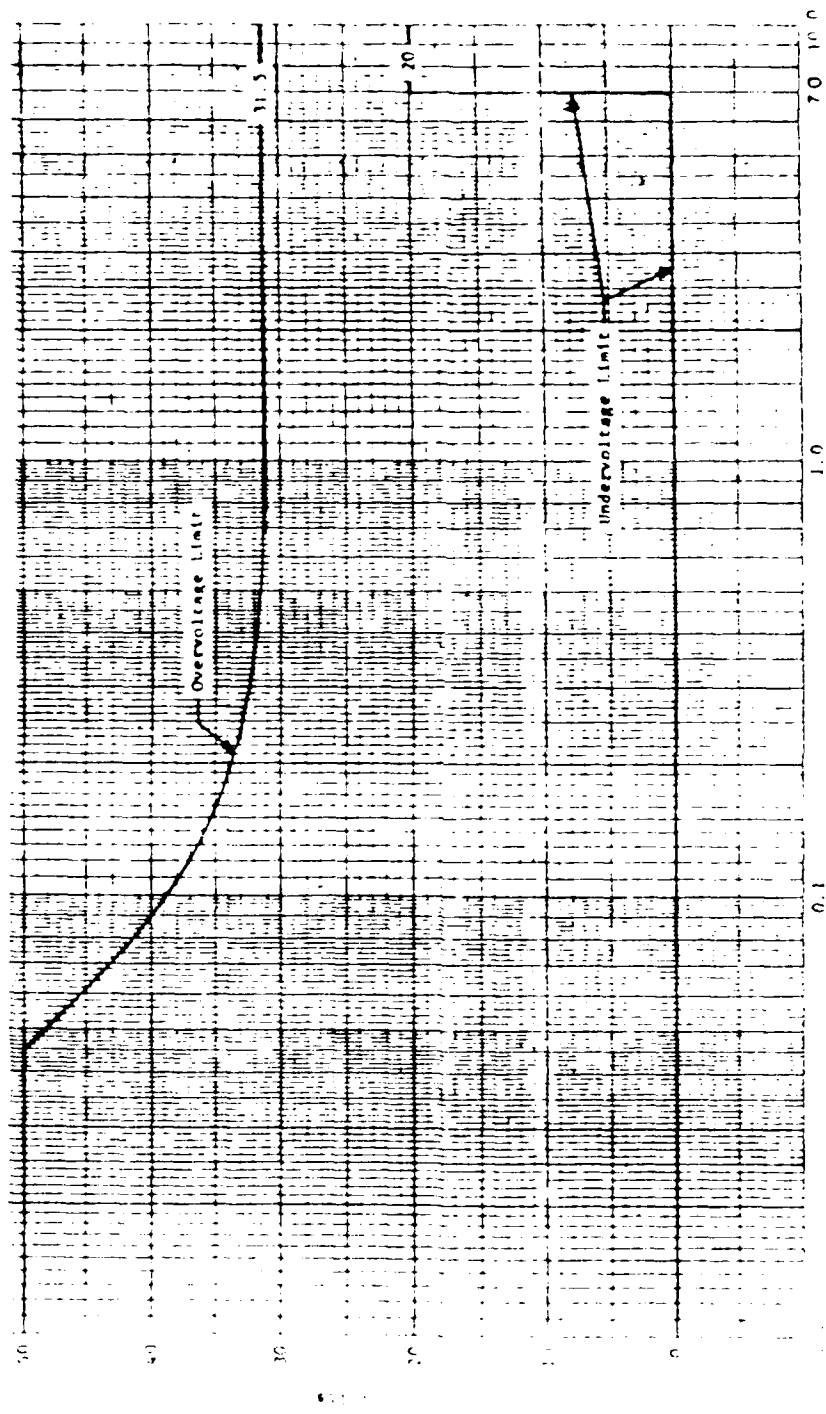
Stability augmentation, flight control, and engine control are particularly needed when the pilot or flight crew are attending to an emergency condition in another aircraft system. Such emergencies are likely to be correlated with abnormal operation of the electric power and cooling systems. The interfaces between flight-critical components and the utilities must be designed to ensure continued operation of critical systems even under abnormal states of the utilities.

The hydraulic and pneumatic utilities usually do not interface directly with the digital components but are important for the input and output devices utilized by the digital functions. The primary safety consideration is that the digital components must be informed of the state of the hydraulic and pneumatic utilities connected to their input and output devices so that appropriate isolation and control gain changes can be effected to compensate for failures in these utilities.



Time from Onset of Overvoltage or Undervoltage, Seconds

FIGURE 4 - 1 AC VOLTAGE LIMITS FROM MIL-STD-704D



Time Following Onset of Overvoltage or Undervoltage, Seconds

FIGURE 4 - 2 DC VOLTAGE LIMITS FROM MIL-STD-704D

#### 4.4.2 Interfaces on Dedicated Links

Dedicated links permit close control of the interface between devices but are expensive to implement and cannot be universally employed in today's highly instrumented and crowded aircraft. A dedicated link may be justified by high bandwidth communication requirements, by the need for extreme timeliness of the information (bus access usually involves some delay), or by security reasons.

Factors to consider in dedicated links are:

- overvoltage protection -- it is undesirable to have supply voltages and digital data on the same connector;
- EMI protection -- typically provided by shielding;
- Definition of wave form and timing;
- Acknowledge/not acknowledge protocols -- this implies bidirectional transmission; and
- Word format, sequence, and error control (detection or correction).

For application to flight critical systems, the consequences of failures at the transmitting end and in the link itself must be considered. The possibility of damage to either end due to high voltages or induced spikes is being minimized by the first two items in the above listing. Functional failures in the originator or link are being made detectable by the latter three items. The handling of a transient failure, once it has been detected, is application dependent. It may involve sending a NAK (negative or non-acknowledge) which prompts the originator to retransmit, or it may just cause the transmission to be ignored and previous values to be used until a valid update is received. Permanent failures can be tolerated only if there are redundant transmitting elements and redundant links. However, temporary failures at digital interfaces are much more common than permanent ones, and a significant effort for tolerating temporary failures is therefore warranted.

#### 4.4.3 Bus Interfaces

Digital buses provide two significant advantages over dedicated links in airborne applications:

- considerable savings in aircraft wiring and connectors (both on equipment and installed wiring); and
- flexibility with regard to adding, deleting or modifying connected equipment.

Two distinct types of buses are in common usage: unidirectional and bidirectional. As the name implies, unidirectional buses permit transmission in only one direction, and they are primarily used for transmitting data from a single source to multiple users. A prominent example is the Mark 33 Digital

Information Transfer System (DITS), defined in ARINC Specification 429, which is used to feed information from one instrument, control panel, or other data source to a number of equipments which use this information.

Unidirectional buses may be used within the flight control system, e. g., for transmission of digital outputs to actuators. Bidirectional buses may also be involved in the transmission of information restricted to the flight control system although this option is not frequently encountered.

The most pertinent bidirectional data bus is that defined in MIL-STD-1553B which is extensively used in Air Force aircraft. It permits interchange of information from multiple sources to multiple users. A significant advantage of the bidirectional bus is that it permits acknowledgement of transmissions and this in turn facilitates keeping back-up information available until the receipt of updates is confirmed. In principle, facilities for acknowledgement can be provided in unidirectional applications by means of multiple buses, an awkward arrangement. ARINC Specification 429 states "No applications for this system capability (acknowledgement) have yet been identified, and thus no ... standards have been established". The primary disadvantage of the bidirectional bus is the considerably greater cost that is associated with its implementation. In some Air Force applications, particularly in transport aircraft, both unidirectional and bidirectional buses are used. A suggested method for interfacing these is discussed in Appendix 2 of ARINC Specification 429.

All precautions for interfacing flight-critical systems by means of dedicated links that were mentioned in the previous subsection also apply to buses. In addition, these are subject to the following failure modes:

- address errors,
- internal inconsistencies,
- denial of access, and
- "babbling".

Each of these failure modes is briefly described below, and suitable protective or fault tolerance measures are discussed.

#### Address Errors

The Mark 33 DITS uses a source address which in effect identifies the nature of the information being transmitted. Any errors introduced in transmitting or receiving this address can cause misinterpretation of the data. This is not a very common failure mode but it can have potentially very serious consequences. A limited amount of protection can be achieved by grouping the input connections to the receiving equipment by "ports", and to allow each port to receive only its specific authorized addresses. Any failures which result in an unauthorized address are thus made detectable.

Bidirectional buses require source and destination addresses, and they are therefore subject to errors in both of these. On the other hand, they permit

the use of acknowledgement procedures which make detection of this failure mode relatively easy.

Data reasonableness tests incorporated in the using routines provide further protection against errors for both uni- and bidirectional transmissions. Alternate data sources must be available to permit operation of flight critical functions after a permanent failure involving addresses has occurred. Temporary and intermittent failures can be tolerated by use of default data or retransmission. It is highly desirable to log the incidence of address failures to permit corrective maintenance to be accomplished at the earliest possible time.

### Internal Inconsistencies

Information transmitted over a bus is constrained to a fixed format, and adherence to that format is frequently checked by the receiver. Internal inconsistencies that are detected in this procedure will result in rejection of the message. Where such checking is not implemented, invalid information might be accepted. A typical MIL-STD-1553B transmission consists of a command word, a status word, several data words, and possibly a final status word. It is seen that interpretation of a status word as containing data, or vice versa, can lead to serious system difficulties. Further, within each word type certain format conventions apply, and violation of these can also cause rejection or misinterpretation of a message.

Most transmission formats include provisions for parity checks on each word or block. Failure of the parity check is a relatively frequent cause of message rejection. Possible responses to a parity error are to request retransmission, to use a default data value, or to use an alternate data source.

### Denial of Access

Most of the unidirectional buses used in current aircraft transmit data from a single source to multiple destinations in a broadcast mode. Since only the source is equipped for transmission and has continuous access to the medium there is little likelihood that a denial of access failure will occur. In some modifications of the Mark 33 protocol alternate sources can transmit on the bus under control of the primary source. In that case failure of the primary device to authorize a desirable or required transmission by the alternate can be construed as access denial.

Denial of access is a more serious failure mode in bidirectional buses. Three fundamental techniques are used for granting access: control by a master, token passing, and contention. MIL-STD-1553B is based on the first of these. The other techniques are currently in use in local area networks and may in the future be applied to aircraft systems. Many malfunctions in the master can cause the bus control sequence to be interrupted and altered, thus denying access to some, and possibly all, users. An alternate master can be activated,

but there is some risk in that procedure, and even if successful it may require a longer time interval than can be tolerated in some flight-critical applications. The preferred means of protecting against access denial is to make the entire bus redundant, and to employ more than one master. This involves considerable expense but is an effective means for dealing with a broad spectrum of bus and interface malfunctions.

The token passing and contention methods do not require a dedicated master for bus access control. In token passing, a user who has temporary control of the bus passes this control (by means of an abstract "token") to another user who can be predefined or who is identified on the basis of computed conditions. The protocol can provide means of circumventing a user who has sustained a failure or who is temporarily ineligible to have access to the bus (e. g., because of low priority). In the contention method a user who needs access to the bus must first listen to determine a quiet period. The user then transmits its identity in a series of ones and zeros. If another user attempts transmission at the same time (this is the contention case), the unit with the greater number of leading ones gains control of the bus. Both token passing and contention are intended to provide continued bus access in the presence of failures in individual units. However, because of failure modes other than access denial, redundancy of the buses may still be desirable in critical applications. The contention method does provide an upper limit on the waiting period before access is achieved. This presents a problem in flight critical systems that require a high frequency response.

#### Babbling nodes

A bus access point which transmits signals that do not conform to the established protocol is called a "babbling node". The most bothersome aspect of this type of failure is that it denies bus access to legitimate users, with a potential of complete disruption of system communications. Failures of this type have been observed due to intermittent connections, electromagnetic interference, and software problems. Diagnostic routines that remove power from selected units are helpful in pointing to the source of the failure. However, only redundant buses can provide assurance that such failures can be tolerated without affecting flight-critical functions.

#### 4.4.4 Software Interfaces

Any kind of software failure can affect data that are being passed across interfaces to flight critical functions. However, under this heading we are particularly concerned about failure mechanisms at the interface proper. Data can be passed from one function to another in three forms: as a message, as a parameter, or through a common data base. The possibility of an undetected corruption of the data generally increases in the order listed.

When data are passed as messages the data value is usually protected by a parity code (mandatory in MIL-STD-1553). Frequently a time tag is associated with the data, and the status of the originating unit or function is identified (e. g.,



operating in normal/restricted/emergency mode). A requirement for this information is established in the message protocol.

When data are passed as parameters (i. e., in the call of one software function to another one), the same information can be provided but this is at the discretion of the software designer and not governed by a protocol. However, in the case of both messages and parameters the originating function is aware of the identity of the recipient and of the use to which the information will be put. This permits tailoring of data and transmission formats.

When information is put into a common data base each data value can usually be accessed by itself. Thus, even if the originator associates a time tag and equipment status with the data proper, there is no assurance that a user will obtain the additional information. During an initial design cross reference lists are frequently generated which permit determination of which functions utilize a given data item. But as modifications are being made, this documentation is not always kept updated, and in a fielded system it is rarely possible to know where and how each item in a common data base is used. The passing of information through a common data base should therefore be used in flight-critical systems only for extremely well-defined data items, and the association of the basic information with fields that indicate its validity should be enforced.

Some examples of how problems can arise are sketched below. A common factor in these examples is that the information processed by the digital equipment is within the range of normal values and is not likely to be rejected by reasonableness tests and similar techniques of broad coverage error detection. This emphasizes the value of formal protocols for passing information between functions.

#### Rate Derivation from a Default Value

A module computing true airspeed has not received input data at the time it started processing. According to the specification, it puts out the previously computed value with a flag that identifies it as a default. Another module computes longitudinal acceleration by differencing true airspeed data. It utilizes the default value and of necessity computes zero acceleration for the latest interval, regardless of the true value of that quantity. The default flag should have been utilized to suppress or alter the computation of longitudinal acceleration.

#### Rate Derivation from a Substituted Value

As above, but the module computing true airspeed accesses an alternate air data source which has a small offset from the primary source. The error in the true airspeed value is tolerable. However, the acceleration computed by differencing the prior value (derived from the primary source) with the current one (derived from the alternate source) is grossly in error. A flag should have been set to

indicate the transition from one data source to another.

#### Change in Filter Algorithm not Propagated

Height above terrain is computed from radar information using a Kalman filter algorithm which requires considerable computer time but provides an accurate and smooth output. When computer idle time falls below some threshold, indicating a possible saturation of the processor, the filter algorithm is modified so that it requires only one-half of the normal processing time. The resulting output is tolerable over most types of terrain, but over some regularly spaced hills it causes the autopilot to command hard-over up to hard-over down elevator (at the natural frequency of the system), endangering the aircraft and crew. The change in spectral composition of the height signal should be communicated to using modules, enabling these to change gain, time constants, or limits.

It will be recognized that in all of these cases the problems arose from an exceptional condition in the mainline data processing, and from failure to communicate to user processes that the exceptional state exists. This is a very common failure mechanism in both ground based and vehicular control systems.

### 4.5 CRITICALITY OF SENSOR AND ACTUATOR INTERFACES

The communications and data aspects of sensor and actuator interfaces do not differ significantly from those described for general interfaces in the preceding section. However, in other areas sensors and actuators pose specific problems because they are an integral part of the flight or engine control systems. Two specific topics of interest with respect to criticality are the detection of failures in sensors and actuators by means of the digital processing, and possible responses to sensor and actuator failures.

#### 4.5.1 Detection of Sensor and Actuator Failures

The speed with which computation and comparison can be carried out in digital equipment provides an efficient and usually effective means of diagnosing incipient failures in associated electromechanical devices. The resulting information can be used by maintenance personnel to replace or repair the affected component, hopefully before a system level malfunction has occurred.

Sensor signals to a digital system are frequently 'conditioned' as a first step in processing. This conditioning may involve static and dynamic calibration, smoothing, and range checking. Where several sensors of a given type are input to the same processor, averaging or midvalue selection are also carried out prior to processing. Many incipient failure modes have specific signatures that can be detected as part of the signal conditioning. Sensor bearing failures or other causes of high friction usually result in a noisy sensor output, and that

is detected in terms of a large value of the sum of squares data that is computed by most smoothing algorithms. Noise due to an incipient sensor failure can be distinguished from that due to a noisy data condition because the latter usually lasts only a short period of time. Some failures that result in a low sensor gain (e. g., leaking bellows in an air data sensor) will cause the sum of squares to be consistently low, and this fact is also a useful diagnostic.

Transducer failures usually occur without advance warning. However, in practically all cases they produce either zero or hard-over output. The latter is easily diagnosed by a range check (and transducer failures are the most common cause of exceeding the specified range), while zero output failures produce a persistent zero value for the sum of squares, again a very unique signature. Transducer type failure mechanisms are also found in many types of tachometers.

Comparison, particularly where three or more instruments of a given type are present, is a broadly applicable technique for detecting incipient and well as actual sensor failures. Techniques for identification of the malfunctioning unit include:

- deviation from average or mid-value, either in absolute terms or as a fraction of the normative reading;
- deviation from average sum of squares (or other measure of sensor noise); and
- deviation from average reading-to-reading differences (indicative of calibration problems).

Where fewer than three identical sensors are installed, comparison can be carried out with computed values of the sensor data, e. g., attitude rate derived from an inertial platform.

A broadly applicable technique for detecting incipient actuator failures is to monitor the residual error signal of the actuator feedback loop as shown in Figure 4 - 3. The position error signal (E in the figure) should have a large value only when high rates of actuator travel are commanded. Failures of the actuator transducer can be diagnosed in a manner similar to failures in sensor transducers. Failure of the correct feedback signal from an actuator transducer will typically result in a hard-over output, and detection of that condition is therefore desirable to permit protective measures to be initiated.

#### 4.5.2 Response to Actuator and Sensor Failures

The response to an acute actuator failure (i. e., one that is not masked by redundancy provided as part of the immediate function) must be taken at a higher system level. Aircraft with split flight control surfaces represent a comparatively simple case. An actuator failure involving one portion of the surfaces can be responded to by recentering the affected surface where possible (by bypassing the actuator or shutting off the hydraulic supply), and by increasing the system gain (deflection/attitude error) of the remaining channels.

There is increasing interest in utilizing auxiliary surfaces (canards, leading edge slats, etc.) for emergency control in case of complete loss of a primary control function or surface. These techniques are particularly applicable to dealing with battle damage. Because the aircraft are at best only marginally stable under these conditions, and because the control action differs from that which pilots are used to, it is important that digital systems are designed for multiple control modes and can manage both the mode transition and the control of the aircraft in the resulting mode. Details of these techniques are beyond the scope of this Handbook.

Where multiple sensors of a given type are provided, the response to an acute sensor failure is the deletion of the affected sensor from the input data set. This is desirable even where the data from the failed sensor are masked, because any temporary disturbance in the output of the remaining operational sensors might cause the voting or comparison algorithm to produce an unexpected value due to the presence of the data from the faulty sensor. Where direct alternative data sources are not available, suitable emergency input data can frequently be computed from functionally equivalent sensors (e. g., attitude rate derived from an inertial platform, temperature at the engine inlet computed from sensors in adjacent areas).

The transition to and the use of alternate data must be made known to using functions. Note the examples cited in 4.3.4 for problems arising at software interfaces due to lack of knowledge about state changes in associated functions.

#### 4.6 Problems in System Integration

There is an increasing trend toward the integration of several digital control functions on Air Force aircraft, particularly tactical fighters. Substantial performance advantages are frequently achieved by the integration of several functions such as weapon sighting, flight control, and engine control. There may also be weight and power savings due to the consolidation of equipment. However, where one of the functions is flight critical and one or more others involved in the integration are not, there is a potential that the entire integrated function will be flight critical. This will necessitate an expansion of reliability activities such as failure modes and effects analysis and system safety analysis, increase in the test effort, and corresponding additions to the operation and maintenance cost for inspection, record keeping, and parts control. Even where all of the functions to be integrated are flight critical by themselves, the increase in interactions resulting from the integration may cause a substantial increase in the safety and control efforts.

A key to the impact of integration on safety aspects resides in the steps taken to prevent the propagation of failures from one segment of the integrated system to another. Functional integration, such as the utilization of weapon system sight errors in the flight control system or utilization of vertical flight path errors in the engine control system, can usually be handled without undue risks. The export of information from a flight critical system to other functions usually presents no problems at all. The import of information to a critical system requires safeguards at the interfaces (see Section 4 of this chapter). The greatest exposure to safety problems occurs in the physical integration, particularly where:

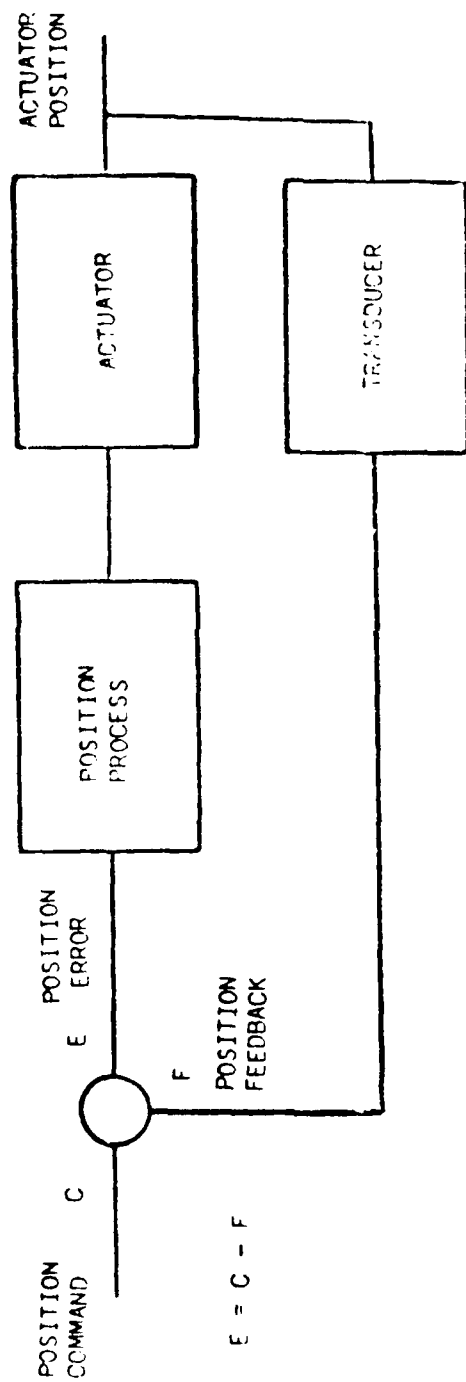


FIGURE 4 - 3 GENERATION OF ACTUATOR POSITION ERROR SIGNAL

- all functions are processed in the same computer,
- functions share a common data base, and
- there is integration of sensors and displays.

The concern with processing in a common computer arises from the capability of any program that is executed to interfere with the execution of other programs due to:

- failure to terminate (endless loops or waiting for message acknowledgement),
- writing into the memory area of a critical program due to a hardware or software failure, and
- competition with critical programs for access to required resources.

While techniques are available for reducing the likelihood of all of these events in normal operation, there remains a possibility of problems under exception conditions, just as the need for continued operation of critical functions is greatest.

Problems that can arise from passing information by means of a common data base have been addressed in Section 4.4.4. The potential for data base errors increases with the size of the data base, and favorable experience on a small data base should not be translated into the expectation that the same circumstances will prevail as the data volume is expanded. There is particular concern about the utilization of large state matrices for flight critical functions. From the control theory point of view the estimation of system states can be improved by increasing the number of observable quantities, which corresponds to increasing the size of the state matrix. The computer workload tends to increase even more than the squared size of the matrix, and the probability of failure increases exponentially with workload (the relation shown in Figure 3-6 for the effect of input/output operations on failure probability also holds for other indices of computer workload). A compromise can frequently be reached by partitioning of the state matrix which may involve a small penalty in the accuracy of the estimation but greatly reduces the demands placed on the computer.

The danger in use of common sensors and displays arises from a reduction in the independent data sources utilized by the digital equipment and the pilot, respectively. Comparison of data sources may be used for detecting actual or incipient failures. The safety value of physical or functional redundancies may not have been fully realized during the analysis that led to the integration.

The greatest need for integration arises from the increase in functional capabilities, such as reduced sight errors in weapon systems, increased performance or reduced fuel consumption for the engine, etc. These objectives can be accomplished without extensive physical integration by using conventional methods of information interchange, preferably by means of messages (see Section 4.4.4). The equipment savings that are made possible by physical integration will frequently be negated by increased expenditures for safety and by the lack of flexibility that is incurred as more and more functions are being integrated

into a single system. The number of functions and agencies that will be involved in an engineering change proposal may be taken as a rough indicator of the time and cost required to implement a change.

## Chapter 5

### TECHNIQUES FOR RELIABILITY, FAULT CONTAINMENT, AND FAULT TOLERANCE

This chapter introduces the reader to techniques and tools that can be used to deal with the hazards to flight critical functions that were described in the preceding chapter. The objectives of the various procedures are presented, and benefits and costs associated with their application to flight critical systems are discussed. Where they are available, military or other Government guidance documents are referenced. The chapter starts with a review of conventional reliability improvement procedures and through the first seven sections deals with increasingly powerful and costly fault tolerance techniques. The final three sections deal with application information common to most of the techniques and with criteria for trade-offs and selection.

#### 5.1 CONVENTIONAL RELIABILITY IMPROVEMENT TECHNIQUES

Practically all current military aircraft system procurements invoke MIL-STD-785 "Reliability Program for Systems and Equipment -- Development and Production" and MIL-Q-9858 "Quality Program Requirements". The former is concerned with reliability issues arising from the application of parts to the equipment and system, while the latter emphasizes issues relating to process control and inspection. Microelectronic devices are subject to MIL-STD-883 "Test methods for Microelectronics". These documents (or more specific requirements derived from them) constitute the baseline of fault avoidance techniques for flight critical systems. The present section emphasizes procedures applicable at the part and assembly level, while system level activities are addressed in Section 2 of this chapter.

Examples of reliability program techniques are derating of parts (e. g., using a capacitor rated for a working voltage of 600 VDC when the actual working voltage will never exceed 300 VDC), and part or assembly screening (e. g., subjecting the articles to a vibration or temperature cycling environment that accelerates the failure mechanism, and removing all that fail or show a significant parameter change). Examples of parts program techniques are calibration of all measuring instruments used in manufacture and test, control over incoming materials, and use of statistical quality control techniques (these may cause entire lots to be rejected when the percentage of defectives exceeds a threshold). The microelectronic test requirements include visual inspection under magnification of samples or of all devices (check for scratches, voids or narrow conductor sections, tritizing and ins dielectrics), tensile strength of lead bonds, and hermeticity of packages. In addition, there are functional requirements which vary with the type of device being tested.



Two questions are very importantly associated with the relation of the cost of reliability practices to fault tolerance:

1. Can reliability, quality control, and test methodology be relaxed when fault tolerance is introduced?
2. Can tighter reliability, quality control, and test practices reduce the need for fault tolerance?

In most cases the answer to both of these questions is in the negative. Military specifications provide an environment for reliability practices within the supplier community which is near the economic optimum at any given time for typical aircraft applications. As indicated in Figure 5-1, the principal variable components of the total cost to be considered in this connection are the cost of failure and the cost of reliability improvement. The basic manufacturing cost does not vary significantly with the failure rate and does not enter into this trade-off. The optimum region is shallow, and total cost is not materially affected by deviations from the exact optimum failure rate. Relaxing the specifications from a level to which industry has already become accustomed will not result in great savings. On the other hand, trying to achieve a significantly lower failure rate than the baseline methodology will incur sharply increasing costs.

Improvements in reliability and parts control are constantly being made, and the cost of reliability curve tends to flatten out, thus moving the optimum point toward lower failure rates. However, at any given time the reliability improvements that are possible by stricter fault avoidance techniques are quite limited. Where an immediate reduction of the system failure rate by a factor of two or more is required, one of the fault tolerance techniques discussed in later sections of this chapter is more likely to produce the desired result at a given resource expenditure.

Fault avoidance techniques do not usually have a significant impact on the size, weight, and power requirements of the equipment. This is their greatest advantage over fault tolerance which usually requires the addition of some components. Conventional reliability techniques have no effect on single point failure modes (but they can affect the probability of failures due to these modes) and they are not usually employed for compliance with MIL-F-9490 or equivalent specifications.

Where fault tolerance techniques are employed, a very significant benefit of fault avoidance is the reduction of maintenance requirements. Parts which do not fail do not need to be replaced. Fault tolerant equipment also interfaces with non-fault tolerant equipment, and these interfaces are quite sensitive to the level of reliability that is being practiced in the non-fault tolerant area. An often overlooked interface of this type exists between the fault tolerant system and its test equipment (which practically never incorporates fault tolerance). Frequent failures in test equipment may leave the fault tolerant system unserviceable or impede its integrity. Conscientious and consistent application of good conventional reliability and parts control practices to support equipment for fault tolerant systems is highly desirable.

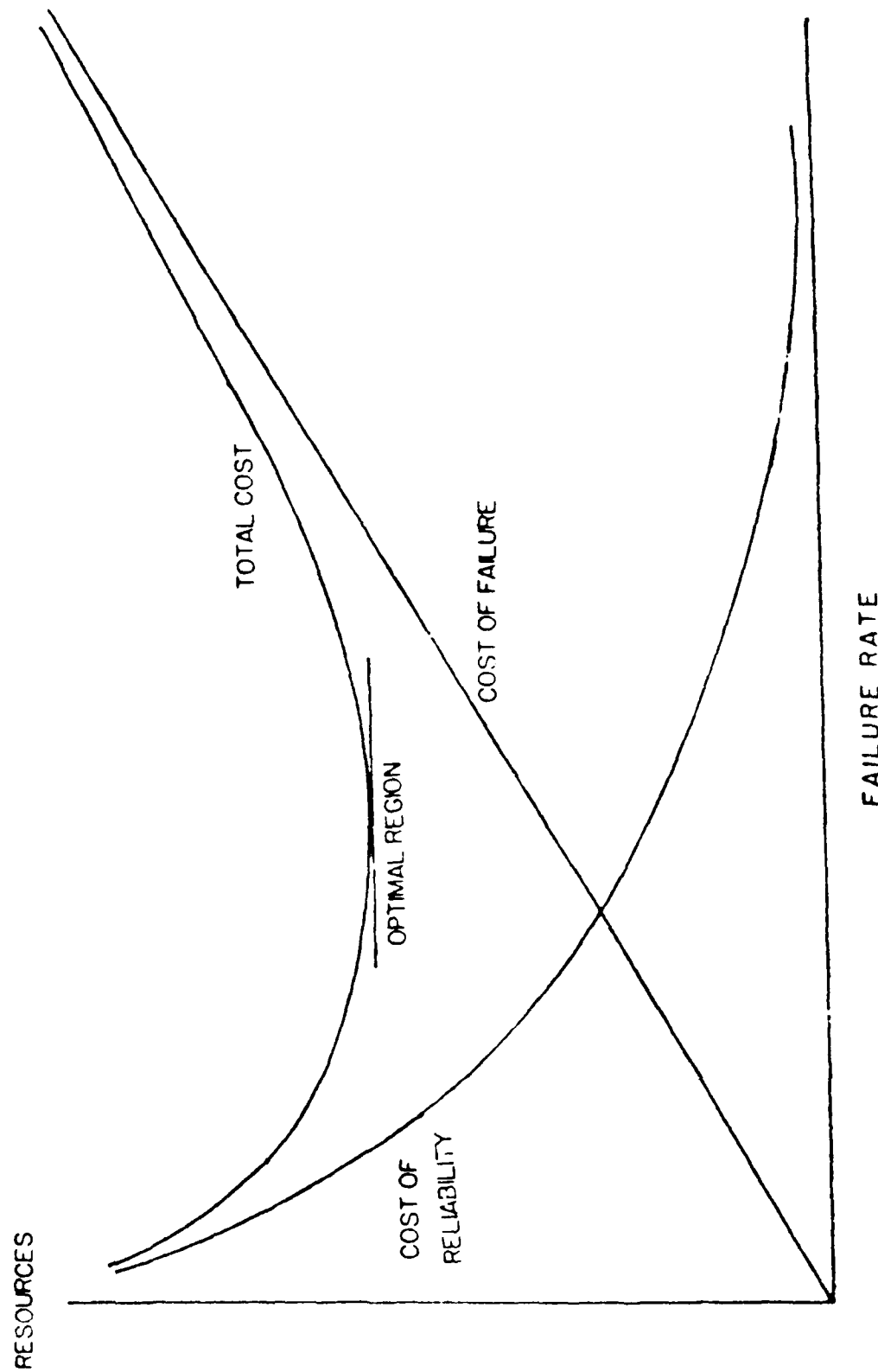


FIGURE 4.1 ECONOMICALLY OPTIMAL RELIABILITY

## 5.2 SYSTEM ORIENTED RELIABILITY IMPROVEMENT

This section examines a number of system level techniques for increasing reliability and for reducing the impact of failures on system capabilities. This is a representative rather than exhaustive description of available methods. All of the procedures employ analysis to identify weak spots in the system reliability chain. The corrective action can involve redesign, employment of local redundancy (e. g., using relay contacts in parallel to ensure closure of a critical circuit), or fault tolerance of a wider scope. Because the latter is discussed in later sections of this chapter the other techniques will be emphasized here.

Three analytical procedures of broad applicability and a fourth one specifically aimed at software are described in the following:

1. Failure mode, effects and criticality analysis;
2. Sneak circuit analysis;
3. Fault tree analysis; and
4. Dynamic analysis of software.

For optimum coverage several of these techniques can be combined, and there are some proprietary methodologies which apply such combinations to a specific equipment area.

### 5.2.1 Failure Mode, Effects, and Criticality Analysis (FMECA)

For U. S. Air Force applications, the conduct of a FMECA is governed by MIL-STD-1629. The standard permits the analysis to be conducted at several equipment levels (referred to as "indenture levels" in the document). For the purpose of this Handbook the lowest level will be that at which the criticality of failure can be assessed. This is generally the Line Replaceable Unit (LRU) but in the case of control panels and wiring a breakdown to individual switches, connectors, etc. can be desirable. Where the LRU is a digital component, hardware and software effects can and should be considered together in the FMECA. It is sometimes believed that software failures are implicitly covered by the analysis of hardware effects because equipment failures are manifest only at the hardware level (e. g., an improper signal transmitted). This reasoning does not account for correlation of hardware failures due to faulty software (e. g., multiple gates being improperly turned on) or for periodicity of failures that cannot be predicted at the hardware level (e. g., a failure associated with a software counter overflow which occurs at the period of an aircraft flexure mode and can thereby produce catastrophic effects).

The FMECA is conducted in two steps:

- Failure Modes and Effects Analysis (FMEA)
- Criticality Analysis (CA)

The scope of the analysis to be conducted in each step is shown in the MIL-STD-1629 worksheet formats which are reproduced in Figure 5-2. The overall purpose of the procedure is to identify those failures which produce severe effects and which have a high probability of occurrence. A graphical representation of this method is shown in Figure 5-3 which is taken from Figure 102.2 of the standard.

MIL-STD-1629 permits the CA to be conducted in either a qualitative or a quantitative manner. In the former, the probability of failure in a specific mode is determined subjectively. Five probability levels are used in this approach, ranging from "frequent" (greater than 0.2 over the item operating interval) to "extremely unlikely" (less than 0.001 over the item operating interval). The value of the data obtained is obviously dependent on the objectivity and knowledge of the person who makes this assessment. In the quantitative CA methodology the probability of failure is determined with the aid of MIL-HDBK-217. The difficulty with this approach is that MIL-HDBK-217 lists failure rate data at the part level. While the overall failure rate of an LRU can be computed fairly readily from the part failure rate information, the failure probability in a specific mode pertinent to the aircraft level depends almost entirely on judgment. Thus, a considerable subjective component enters into this approach as well.

A further problem that sometimes affects the usefulness of FMECA in connection with flight critical functions is that personnel who perform the FMECA tend to be equipment specialists who may not be able to assess effects at the aircraft level. Thus, a transient deviation in an altimeter output may be regarded as being a low severity failure whereas the propagation of this anomaly through the flight control system may result in a much more severe effect.

Benefits of a well conducted FMECA include:

- cataloguing of all failure modes of equipment that constitutes the flight critical system or furnishes direct access data to that system;
- a preliminary listing of failure effects at the aircraft level (to be refined by use of other analyses); and
- a preliminary identification of failure modes that produce severe effects and also have a high probability of occurrence.

Because the FMECA is a bottoms-up procedure (equipment level failures are propagated to the system level) it identifies problems in a manner that frequently permit resolving them through redesign or other measures short of major redundancy or fault tolerance. Examples of actions that might result from an FMECA are:

- inverting the voltage level of an alarm output such that there is an explicit alarm condition on failure in the equipment power supply;
- initiating an immediate reset of the affected function in case of a

SYSTEM \_\_\_\_\_  
 PRIORITY LEVEL \_\_\_\_\_  
 REFERENCE DRAWING \_\_\_\_\_  
 NO. 10-000 \_\_\_\_\_

DATE \_\_\_\_\_  
SHEET \_\_\_\_\_  
COMPILED BY \_\_\_\_\_  
APPROVED BY \_\_\_\_\_

[illegible]

### CRITICAL ANALYSIS

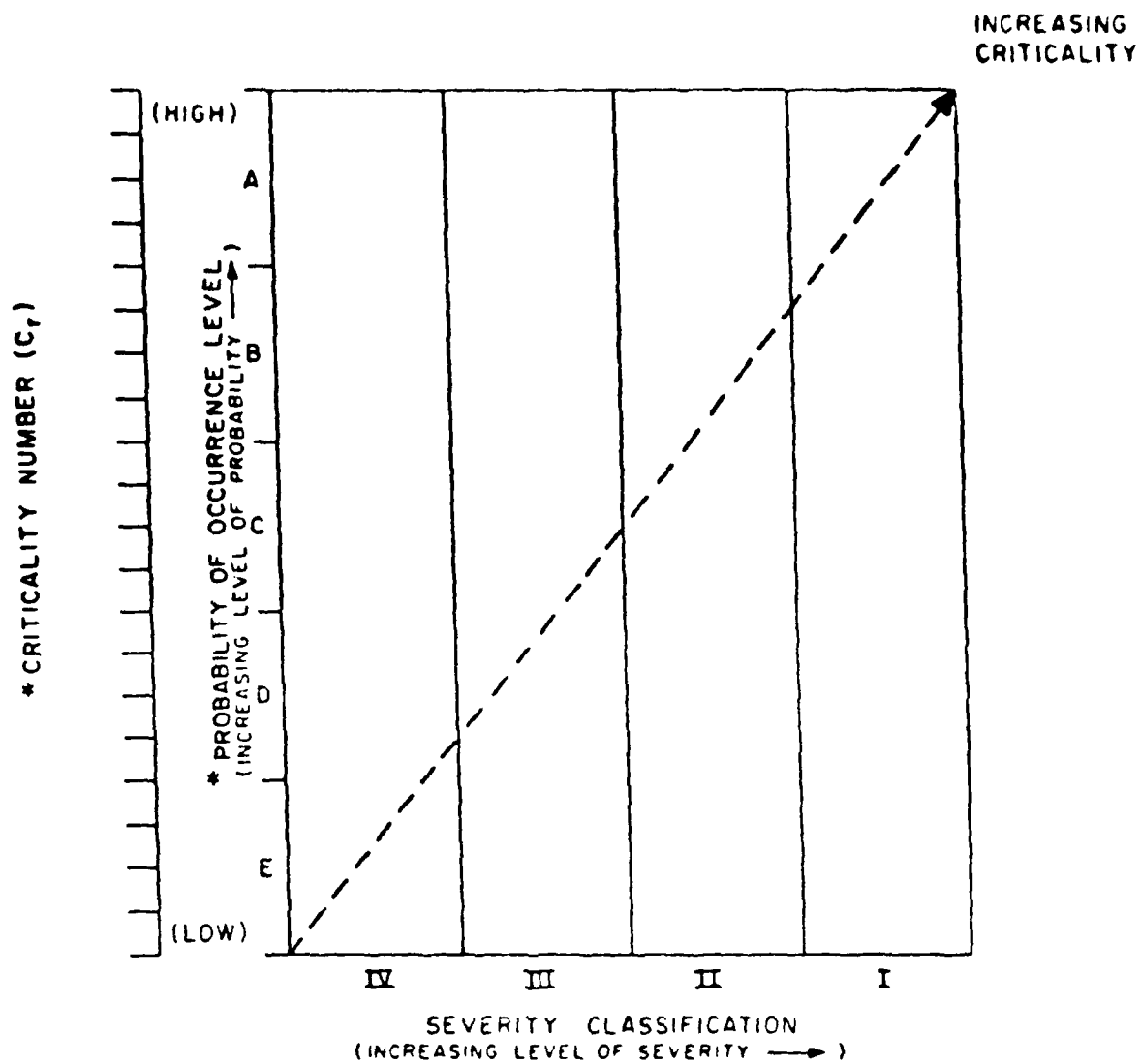
04170 \_\_\_\_\_  
1001000000 1001 \_\_\_\_\_  
001010000 000000 \_\_\_\_\_  
0 1000 \_\_\_\_\_

DATE \_\_\_\_\_  
 BY \_\_\_\_\_  
 COMPILED BY \_\_\_\_\_  
 APPROVED BY \_\_\_\_\_

[illegible]

### B. CA Worksheet

FIGURE 5-2. FMECA WORKSHEETS



\* NOTE: BOTH CRITICALITY NUMBER ( $C_p$ ) AND PROBABILITY OF OCCURRENCE LEVEL ARE SHOWN FOR CONVENIENCE.

FIGURE 5-3 EXAMPLE OF CRITICALITY MATRIX

temporary failure (i.e., eliminating the need for a higher level of redundancy to cover this event);

- separation of power and signal conductors to reduce the severity of short circuits; and
- requiring periodic test of a function which is seldom exercised during routine operation.

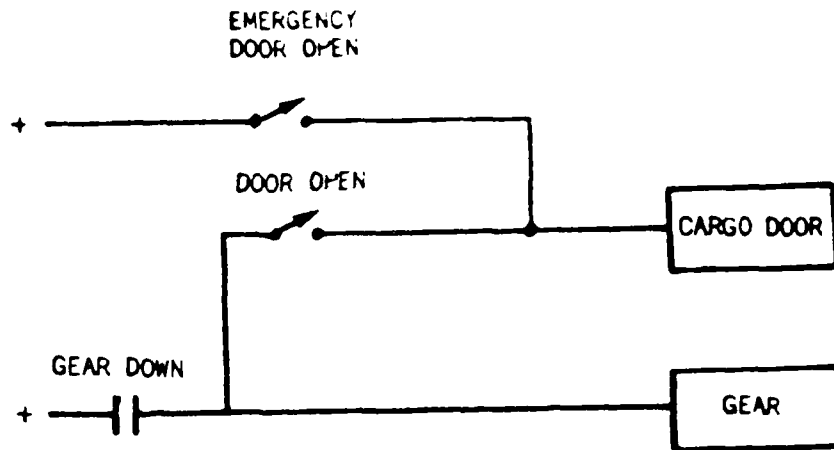
All of these measures can be accomplished at a low cost if the need for them is defined early in the program life cycle. MIL-STD-1629 states "The FMECA shall be initiated early in the design phase to aid in the evaluation of the design and to provide a basis for establishing corrective action priorities." However, an update of the FMECA prior to placing an aircraft type into operation or after an extensive modification program may also be desirable to identify failure modes and effects due to changes (or those that were not recognized during the initial analysis, possibly because there was not sufficient knowledge about LRU characteristics). The FMECA can be extended to include damage effects but treatment of that subject is outside the scope of the present Handbook.

#### 5.2.2 Sneak Circuit Analysis

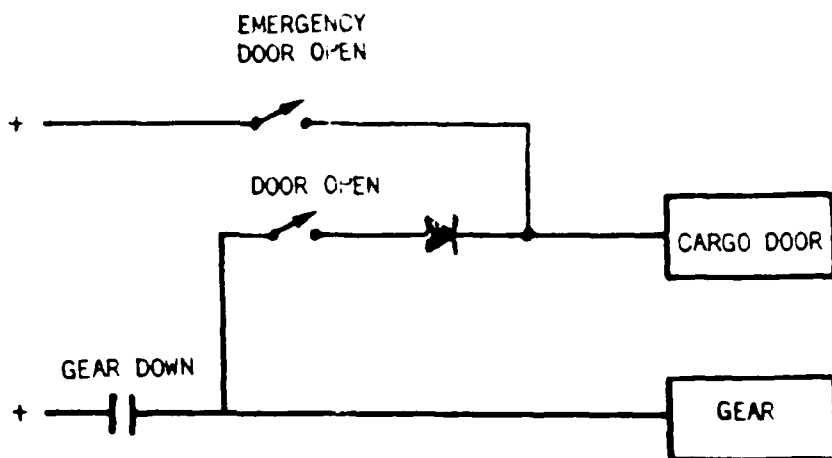
A sneak circuit is a path that can lead to unintended actuation of a device or, conversely, cause unintended deactivation or interference with activation. Sneak circuits typically arise in the logical control and interlock areas. A simple example of a hardware sneak circuit is shown in Figure 5-4A. It is intended to prevent routine opening of a cargo door unless the aircraft is on the ground. For this reason the switch that controls the door opening is energized through the Gear Down contraction. However, there is a requirement for emergency operation of the door when the gear is not down. An independent circuit supplies a safed emergency switch that permits door operation. If the normal door switch is in the closed position, closure of the emergency door switch will cause the landing gear to be lowered. Once this condition is recognized, it can be prevented easily by insertion of a diode into the routine door opening circuit as shown in part B of the figure.

In a practical aircraft system the connections which can cause sneak circuits are vastly more complicated than shown on Figure 5-4. Computer aids for analyzing the control logic are therefore frequently used. After the logic tree (the Boolean equivalent of an interlock schematic like the one shown on Figure 5-4) is entered, the computer produces a printout of all conditions that can cause a given output, all conditions that are necessary to prevent a given output, all conditions that can cause a combination of outputs, etc. After these listings are obtained, it is still necessary for an analyst to review them and to determine whether any of the stated conditions violate the system requirements.

The sneak circuit analysis technique has been extended to computer programs as "Software Sneak Circuit Analysis". Most of this work was performed under NASA sponsorship and many of the computer tools utilized are in the public domain. These tools also perform conventional software structure analysis, check for consistency of variable and label naming, and audit for adherence to program



A. Original Circuit



B. Revised Circuit

FIGURE 5-4. EXAMPLE OF A SNEAK CIRCUIT



design practices, thus overlapping many of the functions described in Section 5.4.4. As in the hardware area, computer based tools are a valuable aid but in the end the evaluation depends on the experience and skill of the analyst.

Hardware and software sneak circuit analysis are significant fault avoidance techniques. They are applicable to control logic, including specifically the control logic that is required in connection with fault tolerance such as computer reconfiguration and software restart.

### 5.2.3 Fault Tree Analysis

In fault tree analysis undesirable events are postulated at the top system level, and then conditions which can cause these events are identified and eliminated. Where complete elimination is not possible, a configuration must be established which brings the probability of occurrence below an acceptable threshold. The primary means of accomplishing this is through fault tolerance. Fault tree analysis can also be applied to the reconfiguration and recovery provisions of fault tolerant systems for the purpose of validating the effectiveness of these provisions.

Fault tree analysis is mentioned in MIL-STD-882 "System Safety Program Requirements" as an acceptable technique but no detailed format of it is specified. The most prominent current use of fault tree analysis occurs in the review and licensing of nuclear power plants, and the Nuclear Regulatory Commission "Fault Tree Handbook", NUREG-0492, is a widely used reference. An example of a fault tree is shown in Figure 5-5. The electronic portion of the pitch control used in that example is assumed to consist of three independent computers and two independent fault isolation networks. Survival of at least one fault isolation network and one computer are required for operation of the electronic portion. Analysis of software failures is indicated as a separate task.

As was the case with the two previously mentioned analytical techniques, fault tree analysis can be conducted more easily with the help of computer based tools. Several tools of this type have been developed for the Nuclear Regulatory Commission and are in the public domain [OLMA82, VESE80]. Proprietary systems are in use in the chemical and process control fields [TAYL80].

Fault tree analysis has also been applied to computer programs [HEFH82, TAYL81]. The first reference is particularly applicable to software for fault tolerant computers. Figure 5-6, taken from that reference, shows the fault tree for the executive program of the Fault Tolerant Multi-Processor (FTMP), a NASA/Draper Laboratory project targeted for flight critical systems [SMITH83]. The procedures used for software fault trees can usually handle combined hardware and software faults. An application of this type to an Air Force armament system is described in [MIL85].

Fault tree analysis is a top-down technique. It should be conducted or supervised by personnel who have a good understanding of critical events at the aircraft and flight controls level. A very important use of fault tree analysis is to translate the top level critical events into events that must be prevented

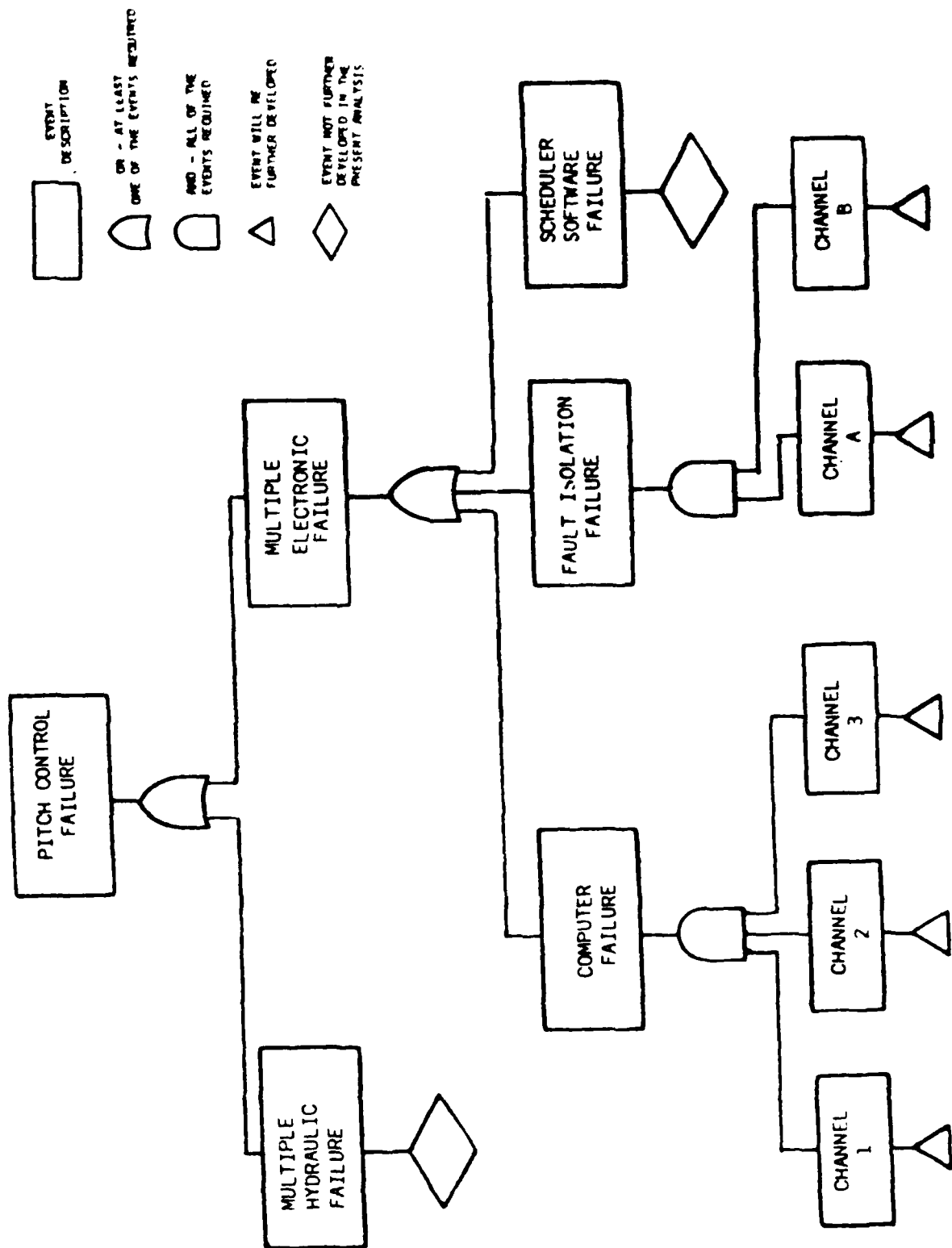


Figure 5-5 Example of Fault Tree

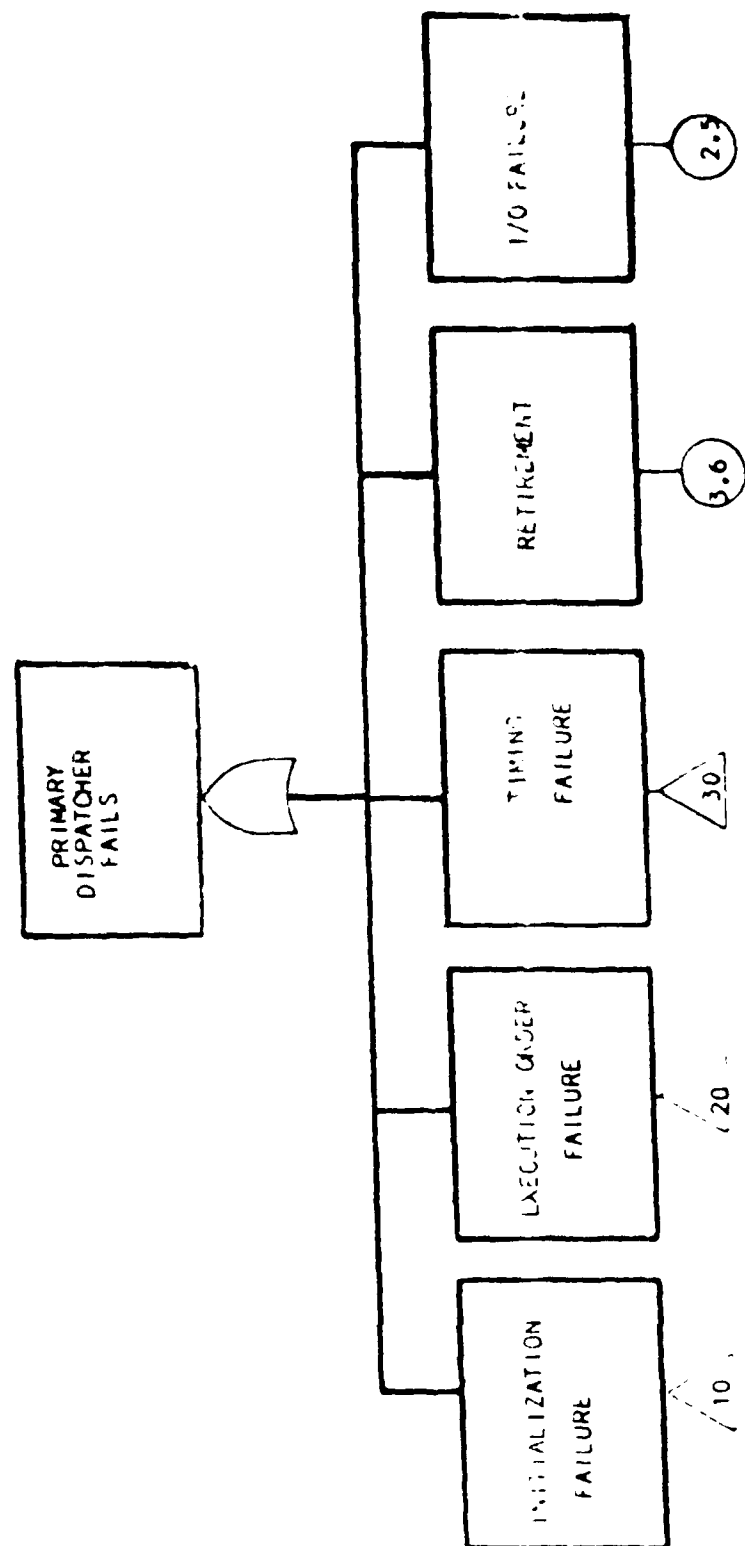


Figure 5 - 6. Example of Software Fault Tree

at the subsystem level. From there on, FMECA can be used to determine the potential causes of the harmful events and to direct their elimination.

Fault tree analysis can also be used in the validation of the reconfiguration and recovery sequence of a fault tolerant subsystem or computer. Key elements in such an analysis are:

- failure of timing or sequencing functions;
- correlated failure in a monitor and in the monitored equipment;
- undetected failures in a monitor or in a spare component; and
- resource contention (e. g., for bus access) during a critical reconfiguration sequence.

#### 5.2.4 Dynamic Analysis of Computer Programs

The term 'dynamic analysis' applied to computer programs means review of the performance of the programs as they are being executed. In a more specialized sense that will be used here, dynamic analysis furnishes a measure of test coverage, i. e., whether the test cases exercise all functions that the program can provide. Since software does not fail due to physical deterioration, the capabilities that are provided at acceptance can be expected to persist throughout the operational period (or at least until a revision is required). If a software function is completely tested and found to perform in accordance with the requirements it can be expected to operate satisfactorily from there on. The obstacle to completely fault-free software is the inability to conduct exhaustive testing on practical computer programs. A valuable characteristic of computer programs is their ability to modify the processing of data depending on some pre-established criteria, e. g., to take one action when a variable is below a threshold and another one when it is above the threshold. This implies that the program may take different branches, depending on the attributes of the input data or computer states. One of several possible measures of test coverage is the ratio of branches traversed during test to the total number of branches [HOWD78]. The dynamic analysis discussed here is aimed at determining the branch coverage ratio and particularly to identify branches that have not been accessed during test. A related metric that is important in modularly structured programs is the ratio of calls executed to possible calls (a call is a transition from one module to another). The assessment of branch accesses and call executions is made with the aid of software tool that are generically referred to as dynamic analyzers or test tools. Publications that provide general background on software tools are [FIPS99, HECH81, HOUG81, HOUG82].

A dynamic analyzer specifically developed for the flight controls field is called AVFS (Automated Verification of Flight Software) [SAIB82]. The performance of dynamic analysis with the aid of AVFS is shown in Figure 5-7. The AVFS commands shown at the upper left identify the sections of the code that are to be subjected to analysis and the specific type of analysis to be conducted. On the basis of these commands the source code (the flight control program) is modified to permit the tracing and counting of branch executions and/or module calls. The resulting program is the instrumented source code

branch within the branch outline which represents the target of the system operation. The audit file is generated during execution and yields coverage reports, some of which are discussed below.

Figure 5-8 shows two reports generated by AVFS. The summary report shown in part A of the figure provides coverage data by module and for the program as a whole. This listing uses "D-D Path" to designate a branch (D-D stands for decision-to-decision). The left column lists the module names and the number of D-D paths in each module. This information is repeated for each test case. The middle panel lists coverage statistics for each module and test case, and the left panel provides cumulative coverage statistics. Note that the cumulative statistics are not the sum of the coverage statistics for the individual test cases since the latter include duplications. By comparing the cumulative statistics for test case 3 and test case 4 it can be seen that the latter provided no additional coverage.

To improve the coverage it is necessary to know which branches have not been traversed so that test cases can be constructed that will access those branches. A valuable aid is the No Hit report shown in Figure 5-8B. For each module the number of paths not hit, and the specific identification of these paths are provided. The major benefit is to know which branches have not been accessed on a cumulative basis. Reports can also be generated that show the frequency of access to branches. This information is useful for performance optimization.

Even complete branch test coverage does not assure that the program will always execute correctly. It may fail due to numerical relationships that were not encountered during test, due to unusual timing conditions, or because of hardware/software interactions. However, dynamic analysis is a practical means of determining that the program has undergone a reasonably thorough test.

### 5.3 FAULT CONTAINMENT

Many types of hardware and software failures produce only transient effects. Regardless of how short the period of abnormal operation, if the contaminated results are permitted to be accessed by later computations they will have permanent effects. The purpose of fault containment is to avoid the use of contaminated data in subsequent processing.

Transient hardware failures can arise from any of the following:

- power supply fluctuations;
- internally generated electro-magnetic interference (EMI);
- externally generated EMI or environmental disturbances (shock, etc.);
- poorly synchronized timing relations or data patterns; or
- natural or man-made radiation.

It has been stated that most hardware failures in current mainframe computers are non-permanent [8, 12, 13]. In well-checked software failures are mostly the

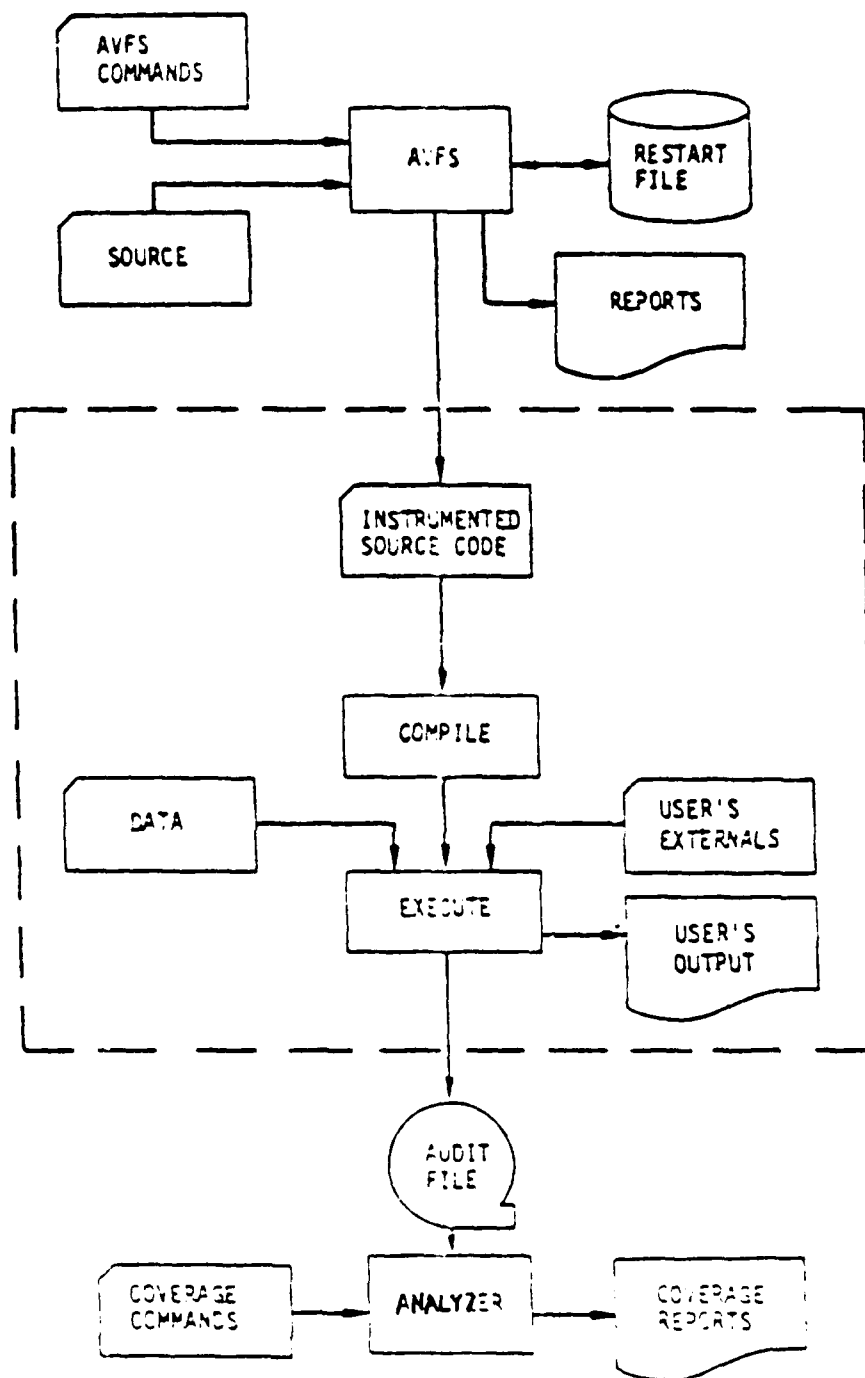


Figure 5 - 7. Dynamic Analysis Procedure



some rare data patterns or coincidence of software execution with computer states (e. g., input/output channel busy). These rare events are not likely to persist and thus will usually result in a transient failure. It is seen that techniques that prevent the progression from temporary to permanent computer failures can play an important part in improving the reliability as seen by the user. It is highly desirable to record the temporary failures (even if the effects are masked) so that underlying causes can be investigated and repaired.

A well-established technique for preventing the contamination of a data base with incorrectly computed values is checkpointing. This involves placing new results into a temporary storage area, called a cache, until they are checked and found to be acceptable. Only at that point are they added to the data base and the cache area is made available for the next batch of computations.

A variant of this technique is the insertion of rollback points in the program. After a rollback point has been passed, no data can be written into the regular memory but data can be read from it. Newly generated data are written into a cache. Before the next rollback point is encountered, all data in the cache are tested, and if any are found to be suspect the program returns (rolls back) to the previous rollback point and re-executes from there. After repeated failures at a rollback point a more severe recovery procedure can be initiated, e. g., program restart. If no failure is encountered when the cache is checked, the content is committed to the regular memory and the program proceeds to the next rollback point.

The effectiveness of these techniques depends on how thoroughly the data in the cache can be checked before being released to the regular memory. The following may be considered as part of the check:

- range of numerical variables;
- increment over prior value for numerical variables;
- sequence (for variables that must either increase or decrease, or for states, such as search, lock, track);
- length and absence of illegal characters in a string variable; and
- comparison with a similar quantity, e. g., left static port pressure with right static port pressure.

Modern computers and computer languages provide some significant error detection mechanisms that can also be brought to bear on the validation of a cache prior to releasing it to memory. Among these are:

- overflow and underflow flags;
- divide by zero alarm;
- recognition of illegal operation codes or memory addresses; and
- type and range checks on variables.

Additional fault isolation procedures can be implemented for messages sent from one computer to another or to a controlled device, providing that the latter is "intelligent" (i. e., can parse the message format). Mechanisms that can be



used to identify correct messages and initiate duplicate transmission for incorrect ones are:

- message serial numbers;
- error detecting code on each character;
- error correcting and detecting code on a message block; and
- end of message signal (to protect against acceptance of partial messages).

Most of these capabilities are provided in the MIL-STD-1553 format which is widely used for data transmission in Air Force systems. To achieve fault isolation, the message is retained by the originator in a buffer (a segregated area of memory) until a valid message acknowledgement has been received from the addressee. If no acknowledgement is received within a selected time, or if a negative acknowledgement is issued (indicating an error in the received copy), the message is pulled out of the buffer and retransmitted, usually with some modification to the serial number so that it can be distinguished from the original copy.

The protection afforded by most message formats is so valuable that it is desirable to use messages to pass information from one process to another even if they both reside on the same computer. The exchange of data through common memory areas does not usually permit safeguards to be applied that are as effective as those associated with messages. On the other hand, the message format requires many more computer operations.

#### 5.4 HARDWARE FAULT TOLERANCE -- CODES AND REPETITION

The techniques discussed in this section provide fault tolerance for some computer functions and a restricted class of faults. In spite of these limitations they are widely used because they require only minimal additions to the hardware and because they are effective against faults that occur fairly frequently.

As the term is used here, codes represent a compressed form of the original information contained in a computer word or group of words. By comparing the compressed form with the recovered instance of the original information some errors can be detected and possibly even corrected. Error detecting code does not by itself constitute a fault tolerance technique and is therefore not discussed under the present heading, although it is a valuable mechanism in support of fault isolation, retry or dynamic fault tolerance.

Error correcting codes provide a complete fault tolerance capability, although of limited scope. The correction capability of the codes described here is due to the fact that in a binary number system each bit can have only two values. If it is known which bit is faulty the value can be reversed and the correct information restored. An error correcting code must therefore locate the exact bit which is faulty, or, in case of a multi-bit correcting code, the exact bits. In most current applications the correction capability of the code is restricted to a single bit, and therein lies one of the limitations of this approach to

fault tolerance. If single bit errors are to be effectively corrected, the design of the functions to be protected by the code, usually the memory and buses, must be such that errors of larger extent are unlikely. This means generally that memory chips that store multiple bits for a single word must be avoided.

Two distinct types of error correcting code are in frequent use: cyclic codes and array codes. An elementary example of each type is presented in the following, but the scope of this Handbook does not permit a comprehensive treatment of coding theory. The reader might want to examine the classic texts in the field, particularly [RA074], or some recent articles [BERL80, MCEL84].

#### 5.4.1 Cyclic Codes

The distinguishing feature of cyclic codes for the applications covered here is that the error correction capability is specific to each word. Thus, errors in the first bit of one word and in the third bit of another word can be corrected even if they occur simultaneously. The example shown in Table 5-1 represents eight information bits to which four code bits have been appended for error correction. Because of the short word length the percentage increase associated with error correction is much greater than in a typical computer word.

The top part of Table 5-1 shows the coverage assignment of each code bit. Thus, in an odd parity convention bit 8 is assigned a value such that the sum of it and the covered information bits (0 - 6) will yield odd parity. Similarly, bit 9 is assigned such that its value and that of the information bits 0 - 3 and 7 will yield odd parity. In this arrangement, an error in any information bit will cause failure of the parity checks of either row 8 or 9 (or both) in the top part of the table. Errors in the code bits as well as in the information bits can be identified.

TABLE 5 - 1 EXAMPLE OF CYCLIC ERROR CORRECTING CODE

Code Bit	Word Bits Covered											
	0	1	2	3	4	5	6	7	8	9	A	B
8	x	x	x	x	x	x	x	x				
9	x	x	x	x					x		x	
A	x	x			x	x		x				x
B	x		x		x		x					x

Error Indication	Word Bit Indicted
8	8
9	9
A	A
B	B
8-9	3
8-A	5
8-B	6
9-A	7
8-9-A	1
8-9-B	2
8-A-B	4
8-9-A-B	0

To locate the error all bits have to be examined, and the error indications involving checkbits shown in the lower part of the table can then be used to identify the faulty bit. It will be noted that not all combinations of error indications are assigned, e. g., the two-bit combinations 9-B and A-B are not used. The four code bits utilized in Table 5-1 can provide error correction for up to 15 bits (including the code bits). Once the location of the error is established, the indicted bit is reversed and the resulting information is accepted as correct.

At this point another limitation of error correcting codes becomes apparent: if the error was due to a temporary disturbance, e. g., an alpha particle hit, the corrected word may be stored back into memory and normal operation resumed. On the other hand, if a permanent hardware fault has occurred, it is undesirable to utilize the affected memory location because correction would be required on every read operation (which slows the computer down), and a second error in that word would not be correctly diagnosed by the code. It is therefore desirable to store the address of the affected word, and to perform repeated store and read operations either as part of the run-time diagnostic program or as part of a maintenance operation. In case a permanent hardware failure is diagnosed, the following alternatives may be selected:

- Continue using the affected address;
- Circumvent this address by program changes;
- Circumvent this address by address translation;
- Use a syndrome store and correct circuit;

- Remove the affected memory block by reconfiguration; or
- Remove the affected memory block by physical replacement.

Continued use of the memory location may be possible if the consequences of data loss can be tolerated. The probability of an independent random error occurring in the same word in military grade memory chips is less than .000001 per read operation. However, very little is known about the probability of correlated faults (e. g., due to environmental factors that might affect more than one memory chip). Circumvention by means of a program change is not feasible in an operational setting but may be used in an experimental aircraft. Circumvention by address translation is a desirable alternative where this feature is available in the computer. Syndrome store and correct circuits are at present used only in a few mainframe computers. Removal by reconfiguration is a practical procedure in many fault tolerant computers. However, an entire block of memory is retired whereas address translation retires only a single location. Physical replacement must ultimately take place to repair the faulty memory assembly. The main objective of the other alternatives is to defer physical replacement until a convenient time.

#### 5.4.2 Array Codes

Array codes can be more efficient than cyclic codes for error correction (in terms of the ratio of information bits to required code bits) but they do not provide coverage for more than a single bit error in a block. They also require more operations when words are written into memory. The hardware necessary for checking array codes is of the same order as that for cyclic codes and is entirely comprised of standard logic functions.

In array coding the location of the error is identified by the intersection of an error detecting code on the word (row) and another error detecting code on the array (column). Each of these can be a single bit parity code. An example of a array code for eight words of eight bits each is shown in Table 5-2. An x represents an information bit, c represents a word parity bit, and b represents a block parity bit. The latter is constructed as the exclusive or (XOR) of all bits in that column.

TABLE 5 - 2    ARRAY CODES

```

x x x x x x x x c
x x x x x x x x c
x x x x x x x x c
x x x x x x x x c
x x x x x x x x c
x x x x x x x x c
x x x x x x x x c
x x x x x x x x c
x x x x x x x x c
b b b b b b b b b

```

In normal read operation only the word parity bits are checked. When a word error is detected, then a block parity check is made. The bit indicated by the block parity is then inverted in the affected word. The procedure can correct only one bit per array at a time, and therefore the blocks should not be made too large. Even for fairly small block sizes (e. g., 128 words), significant savings in storage are achieved compared to cyclic codes.

On writing into memory, the array code has to be updated. This is usually accomplished by (a) forming the XOR of the previous content of the affected word with the new content, and (b) generating the new array code by forming the XOR of the previous one with the result of the operation under (a). Because all computations are of the same type they can be handled by a single dedicated set of registers.

Because array codes are much more advantageous during read than during write operations, they are preferred for program memories in computers where these are separate from data memories. Array codes can be used in computers where the memory is configured with at least one parity bit per word. A major practical use of this type of code is in communications where the words in a message are treated as a block. The term block code is frequently used there.

#### 5.4.3 Repetition

Repetition is redundancy in the time domain. Instead of adding physical resources, some penalty in computer throughput is accepted. This penalty will be incurred only if an error is detected and thus will be quite small on the average.

It can involve retry of elements of an instruction (e. g., a microinstruction), an entire instruction, or an instruction sequence. The most frequent use of repetition is the retry of a memory read operation when an error is signalled by error detecting code. Retries of disk storage (or of other external storage) are also quite common. Retry procedures are implemented in the operating system of many current computers, including some very large computers.

Retry of an instruction sequence is best implemented by means of the rollback technique which was described in Section 5.3.

## 5.5 HARDWARE FAULT TOLERANCE -- REDUNDANCY

The purpose of this section is to acquaint the reader with the basic configurations used in fault tolerance. Practical flight critical systems typically use a combination of several of the basic structures, and a methodology for combining and evaluating these at the system level is described in Section 5.7.

### 5.5.1 Static and Dynamic Fault Tolerance

Two major implementations of fault tolerance are the consensus mechanism and error detection followed by reconfiguration. The most widely employed consensus mechanism is the voter, particularly the two-out-of-three voter used in conjunction with triple modular redundancy (TMR). The voter may suppress an output from computer A in one instance and one from computer B in another without actually changing the connections of the contributing elements. Because no change of the configuration is required, this is termed static fault tolerance. The occurrence of single failures is hidden from users of the information unless specific facilities are incorporated to signal a disagreement among the contributors. For this reason the consensus approach is also called fault-masking. Because it operates without distinction on transient and permanent faults, and because it does not require explicit error detection and switching, the fault masking technique is frequently preferred for flight critical functions that permit only very short interruptions of the computing service.

The distinguishing feature of dynamic redundancy is that errors are explicitly detected and that reconfiguration is required to restore the system to a serviceable condition (failures cleared by retry do not usually activate the reconfiguration mechanism; these can be regarded as fault tolerance at the lower level and as successful operations at the level at which the dynamic redundancy operates). The reconfiguration will interrupt service and the duration of this interruption must be made sufficiently short so that it is tolerable at the system level.

Several interesting variants of these classical types of fault tolerance are of interest. Self-purging logic starts out as a consensus approach employing more than three contributors. If there is no agreement on a vote the disagreeing member is permanently retired (the retirement can be deferred until a number of consecutive disagreements have been registered). The primary benefit of this technique is that a failed computer cannot participate in the vote and therefore the probability of reaching a consensus on a wrong value is reduced [LOSQ75].

Hybrid redundancy also starts out as a consensus system, typically involving triple modular redundancy. When a computer is diagnosed as having failed (this may involve one or more disagreements on voting) it is replaced with a spare. Since this involves both static redundancy and reconfiguration the terminology "Hybrid Redundancy" is quite appropriate [MATH70]. The spare computers can be unpowered until they are configured into the system. This is a major advantage where power or cooling is at a premium. Hybrid redundancy was originally

developed for space applications.

Because the focus of the current discussion is on macroscopic aspects of the redundancy configurations, switches, voters, and comparators are treated as inherently reliable structures. In practice the failure rate of the devices involved may be so small that it can indeed be neglected, or fault tolerant design can be used for these functions. Specific implementation of these configurations are described in the following headings. Reference is made throughout that discussion to Table 5-3. A box without diagonal lines represents an operational computer (a computer that contributes to the system functions). A box with one diagonal line represents a standby (the term hot standby is sometimes used for emphasis), a computer that can be made operational by simply connecting it to the system output devices. A box with two diagonal lines represents a spare (sometimes called a cold standby), a computer that requires at least a memory refresh before it can become operational.

#### 5.5.2 Configurations Employing Two Computers

True consensus systems cannot be constructed with two computers. However, configuration 1 in Table 5-3 provides a limited consensus as long as the two computers agree. Once a failure has been detected, the following actions are possible:

- use diagnostics within each computer to identify the operational one;
- use diagnostics at a higher level (see Section 5.7);
- select a 'safe' output or restrict control authority; or
- select best computer by trial and error (possibly under the direction of a higher system level).

Use of diagnostics within each computer after a failure has been diagnosed by comparison has a high probability of detecting solid faults in the vital areas of the computer. For temporary faults a retry technique can be employed. Solid faults in non-vital areas may not affect routine operations. If diagnostics are unable to identify the faulty computer resumption of operation in the original configuration may be attempted.

Diagnostics at a higher level imply the existence of a systems management or performance monitoring computer which can conduct an independent health check on the flight critical computers. Selection of a 'safe' output is seldom possible in flight critical functions; it is applicable to some engine control functions (see Section 5.8). Restriction of control authority and corresponding restrictions on the aircraft flight envelope represent actions that may be taken together with some other measures. The trial and error approach can be used together with restricted authority to identify the best currently available resources. An rms error criterion for a sensitive quantity (e. g., attitude error) can be used to make the selection.

Configuration 2 consists of an active and a standby computer (a computer that is powered and ready for immediate service). The determination that an error has

occurred in the primary computer is made by internal diagnostics or by diagnostics at a higher level and switching to the alternate is then automatic. The switching logic usually requires a keep-alive signal from the active computer and switches automatically when that is not received. The error detection capability is inferior to that achieved in Configuration 1 but the need for the comparator has been eliminated. Because comparison is usually performed outside of the computer, the information has to be transmitted at a much slower rate than is used for intra-computer transmissions. Where frequent comparisons have to be performed the effect on computer throughput usually represents a much greater penalty than the physical resources involved in the comparator.

The active and spare configuration shown as the last dual system in Table 5-3 functions very similar to the active and standby configuration. The spare need not be powered until required for service. This saves energy and reduces the failure probability but imposes a time lag before service can be resumed following a failure. In most flight critical systems this time lag is not acceptable. However, for large computers which are involved in pattern recognition functions, the energy and failure rate considerations may be important.

#### 5.5.3 Configurations Employing Three Computers

Configuration 4 in Table 5-3, triple redundancy with voting, is the simplest configuration that provides fault masking and is widely used in current flight critical systems. The three computers may run in close synchronism (e. g., from the same clock or from mutually adjusting clocks) in which case the system is said to be tightly coupled. Alternatively, the timing in each computer can be completely autonomous, and that identifies a loosely coupled or asynchronous system. The latter approach is less prone to failures arising from common clock circuits or interfaces, and it may also be less susceptible to EMI upsets (because the information in the three channels is in different stages of processing at the time of the disturbance). However, loose coupling requires more waiting time before a vote is complete, and that usually reduces the computational throughput significantly.

Voters can be made fault tolerant [MCC080]. In many aircraft systems voting is performed at the actuators in devices that are either highly reliable or are redundant and self-disabling in case of a failure. Voting provides near perfect fault coverage for random failures but it is not effective for correlated failures in either hardware or software. Therefore great care must be taken to analyze and remove all possible causes of such failures.

It is highly desirable that the occurrence of frequent or persistent disagreements between channels be logged to permit appropriate maintenance. This requirement adds some complexity to the voter. A further drawback of this configuration is that all three computers contribute to the system failure probability, and that fault tolerance is lost after a single failure. Reconfiguration after a failure (e. g., to Configuration 1 in Table 5-3) can be implemented by special circuits. Three different configurations have to be provided for, depending on which computer failed.



The pair and standby configuration (colloquially referred to as "pair and spare") eliminates the need for a voter and the associated throughput limitations. A comparator (which has a lesser impact on throughput) and a switch (which has no impact until a failure occurs) are substituted. In the simplest case the standby computer is switched in whenever the comparator detects disagreement between the active units. This exhausts the fault tolerance capabilities after the first failure which may be only of a temporary nature. To alleviate this deficiency retries of the active units may be required prior to switching, or a capability for switching back to the pair may be provided.

The standby unit is not required to be of the same type as the active pair, and it may run different software. This capability can be utilized to protect against some types of correlated hardware and software failures. The comparator is not likely to detect correlated failures, and detection capability for that must therefore be provided at a higher level in the system. Configuration 6 employs a single active computer and two standby units (usually called monitors). Because only a single computer is on line at any given time, the fault tolerance provisions have a minimal impact on throughput (the comparison in the monitors does not have to be tightly synchronized with the active computer). In most other respects the features of this configuration are similar to those of Configuration 5.

#### 5.5.4 Configurations Employing Four Computers

The quadruple redundant system shown as Configuration 7 in Table 5-3 is similar in operation to the triple redundant system. The voting algorithm can accept as valid any result for which two identical values have been received. The pathological case in which two computers agree on one result and two on another is decided on the basis of the first agreement that is received. In practice such a case is more likely to arise from a temporary timing problem than from a serious failure. Thus either set of two answers may be accepted (one may pertain to a different time cycle than the other, but both will be within specification). Compared to TMR the quadruple configuration involves additional equipment but it can remain operational after two independent failures. This capability can be utilized for increased safety or for deferral of maintenance after a first failure. Software techniques can be used to retire a computer if its output consistently disagrees with the majority (see self-purging systems in 5.5.1).

The dual-dual configuration shown as entry 8 in Table 5-3 is particularly well suited where the primary aircraft controls employ a dual structure. This can be in the form of split surfaces or two independent actuators operating on a single surface. When a single comparison fails, the affected half of the system can skip output for that computing cycle. When there are repeated failures, the half-system is disabled. In some cases the primary control structure can compensate for the reduction of control authority that follows from the outage of a half-system. The dual-dual configuration can in general tolerate multiple failures only if they affect the same half of the system.

The final entry in Table 5-3 is a hybrid redundant system consisting of a TMR configuration augmented by a single spare. This system remains operational

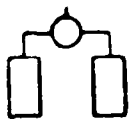
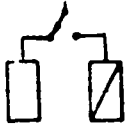
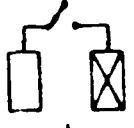
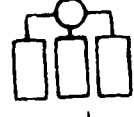
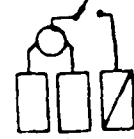

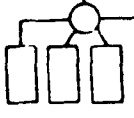
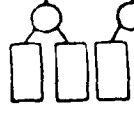
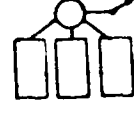



No.	Representation	Primary Error Detection Mechanism	Remarks
1		comparison	best error detection capability in dual configurations
2		diagnostics	eliminates need for comparator
3		diagnostics	power saving and low failure rate for spare
4		consensus	simplest consensus system
5		comparison	comparator and switch substituted for voter; standby may be different type
6		comparison	comparator operates switch
7		consensus	operational after two failures
8		comparison	suitable for dual primary controls
9		consensus	power saving and low failure rate for spare
ACTIVE UNIT  COLD STANDBY  COLD SPARE 			

TABLE 5 - 3 FAULT TOLERANT COMPUTER CONFIGURATIONS

after two independent failures and can be used as a spare for the other computers in Configuration 7. Because the fourth computer is a spare, it requires less operating power and also achieves a somewhat lower failure rate than the quadruple redundant configuration. Because only three computers are active at any one time, the hybrid redundant system can be connected to output devices that incorporate voters for triple commands.

#### 5.5.5 Multiprocessor Systems

In a multiprocessor several processing units access a central memory and share one or more input/output channels. This is generally more efficient than performing the same jobs in several dedicated computers because of the integration made possible by the central memory and because of the sharing of other computer resources. Because flight critical systems must operate under tight time constraints, the increased computational throughput made possible by a multiprocessor system is very attractive. Also, in some cases the multiprocessing capability can be utilized to achieve fault tolerance. Three experimental multiprocessor architectures that have been developed for flight control applications are briefly described below.

Fault-Tolerant Multiprocessor (FTMP) The FTMP has been developed by Draper Laboratory under sponsorship of the NASA Langley Research Center [HARR75, SMIT83]. Essential features are:

- All processing is performed in computer triads, thereby achieving fault tolerance equivalent to the TMR configuration.
- Several triads are active at one time and perform tasks that come from a common queue in the central memory.
- Triads operate from a common fault tolerant clock (tightly coupled).
- There is a central pool of spares from which triads which lose a processor can obtain replacement. When the pool is exhausted, operational processors from non-functioning triads are used as spares.

The general organization of the FTMP is shown in Figure 5-9. A prototype unit is being tested at NASA Langley. The Advanced Information Processing System (AIPS) currently being developed for NASA by Draper Laboratories incorporates many of the features of the FTMP but implements them in a building block fashion that permits implementation of fault tolerant uniprocessors (incorporating either voting or comparison) or multiprocessors [SMIT84]. AIPS is intended for flight critical aircraft functions as well as for space applications.

Software Implemented Fault Tolerance (SIFT) The SIFT concept was generated at SRI International and the development was also sponsored by NASA Langley [WARR75, SMIT84]. Essential features are:

- Individual processors are organized into fault tolerant or non-fault tolerant configurations under software control.
- The degree of fault tolerance can be selected to suit the criticality of

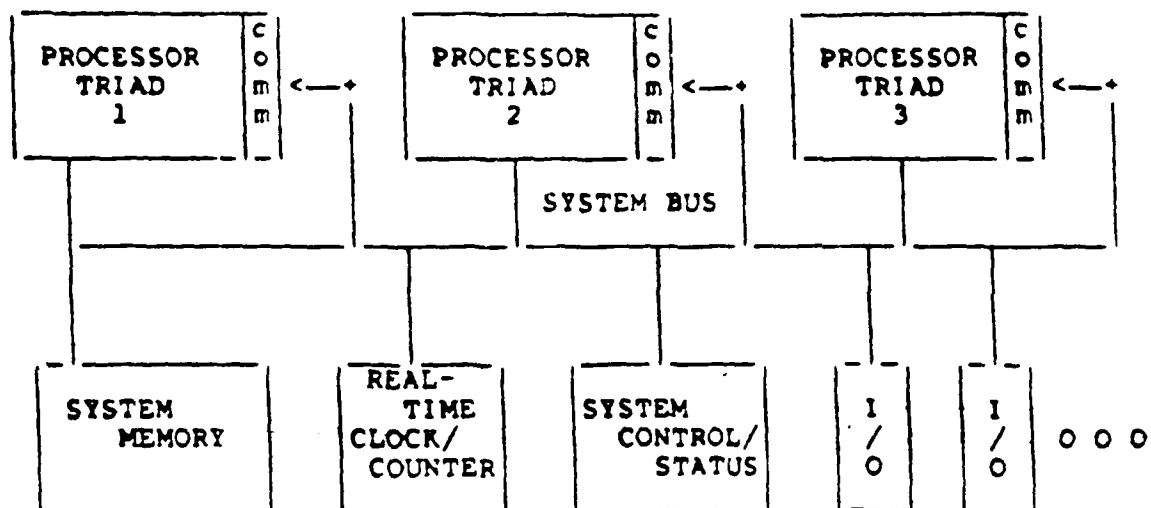


Figure 5-9 General Organization of the FTMP

the task non-critical functions can be performed by a single processor.

- Individual processors are largely autonomous (loose coupling).
- Interprocessor communication is over dedicated links.

The flexibility inherent in this technique is very attractive. However, the dependence on software functions imposes a large throughput penalty. An overview of the SIFT architecture is shown in Figure 5-10. The development unit is being tested at NASA Langley.

Continuously Reconfiguring Multi-Microprocessor (CRMMP) The development of this concept has been carried out in-house at the USAF Flight Dynamic Laboratory [LARI81]. The continuous reconfiguration referred to in the title is accomplished by time slicing. Individual processors alternate in performing tasks during successive time slices. Significant features are:

- Individual processors are largely autonomous but timing is controlled by a central redundant clocks.
- The degree of fault tolerance can be selected to suit the criticality of the task (non-critical functions can be performed by a single processor).
- Spare processors can be utilized for any task (pooled spares).
- There is a virtual implementation of the central memory.

An overview of the architecture is shown in Figure 5-11A. The operation of time slicing is illustrated in part B of that figure. The CRMMP is still undergoing evaluation at the Flight Dynamics Laboratory.

## 5.6 FAULT TOLERANT SOFTWARE

In Section 3 of the present chapter it was stated that many software failures can be overcome by fault containment techniques, such as checkpointing or rollback. There are, however, situations where true fault tolerance is required to recover from a software failure, e. g., where an error has been made in task selection or where the system is 'stuck'. An area that is especially pertinent to flight critical systems is the employment of software fault tolerance techniques for programs that service recovery from hardware failures.

A distinguishing feature of software fault tolerance is the availability of one or more alternate versions of a program which can be used for comparison with the primary program or be executed in its place when the existence of a software error has been determined. The two principal implementation techniques for fault tolerant software are N-version programming and the recovery technique. The former employs redundancy in a manner similar to fault tolerant hardware. The latter depends on explicit detection of errors and the execution of a recovery program which dynamically reconfigures the system.

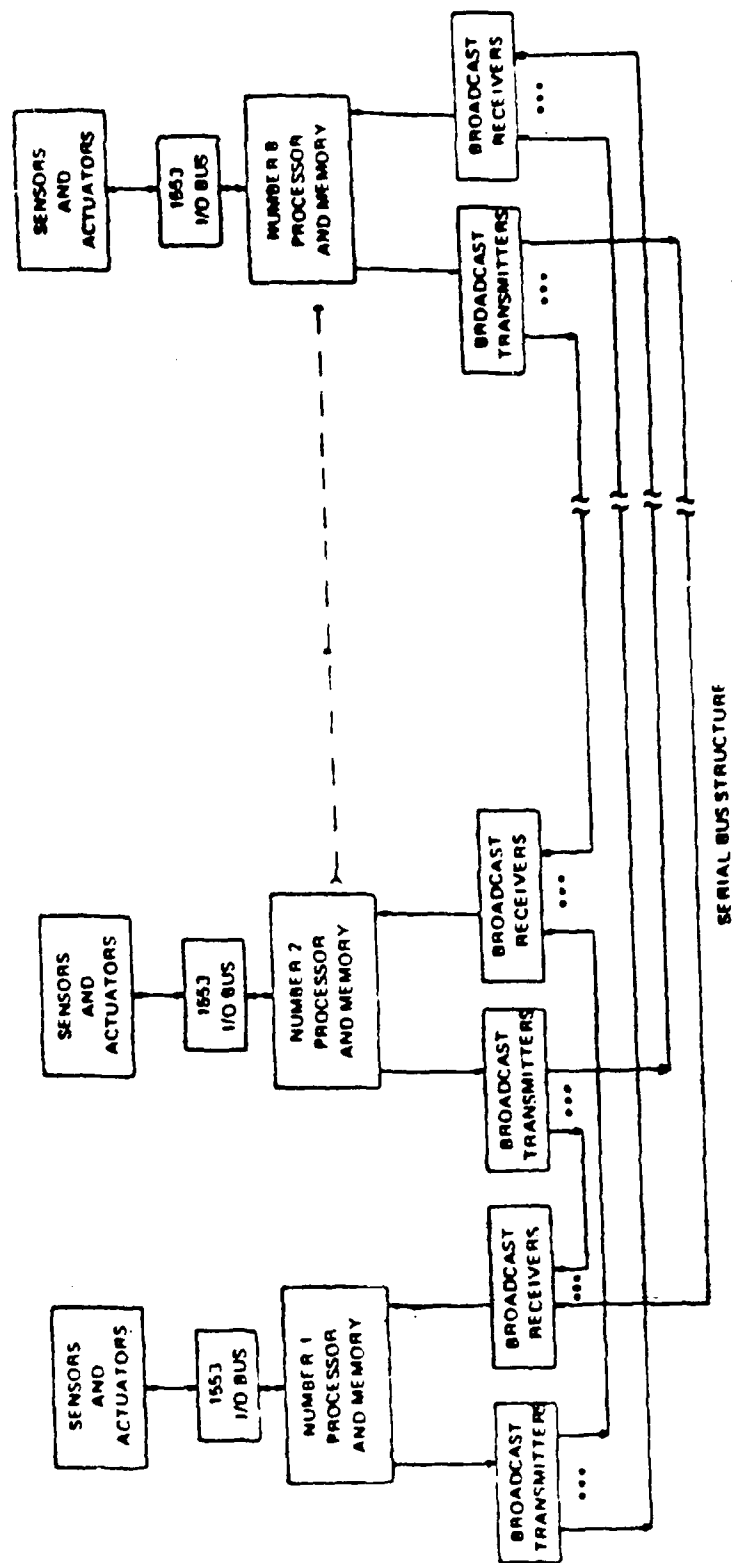


Figure 5 - 10. General Organization of SIFT

COMPUTER RESOURCES HANDBOOK FOR FLIGHT CRITICAL SYSTEMS 2/2

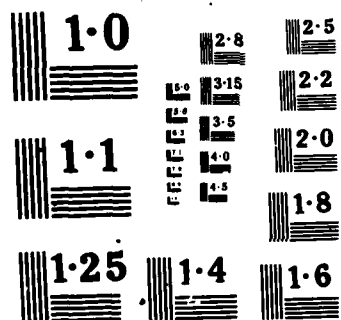
2/2

HECHT ET AL. JAN 85

F/G 1/3

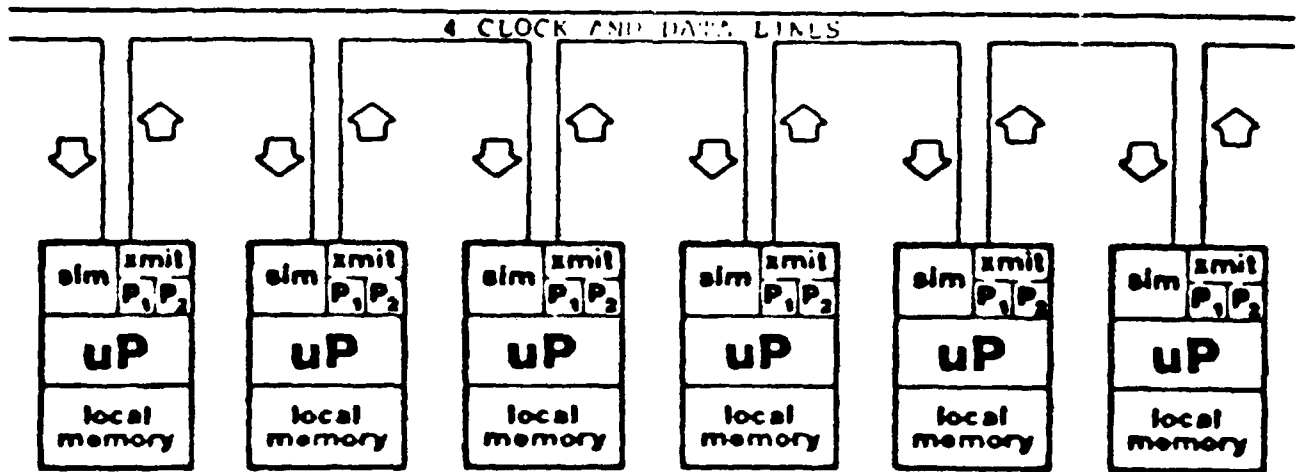
NL

[illegible]



NATIONAL BUREAU OF STANDARDS  
MICROCOPY RESOLUTION TEST





A - Architecture



Time Frame 1



Time Frame 2



Time Frame 3

B - Operation

sim - state of system  
and ty  
... - port 1 & 2

Figure 5 - 11. Architecture & Operation  
of the CHOPS

### 5.6.1 N-Version Programming

In N-version programming a number ( $N - 1$ ) of independently coded programs for a given function are run at the same time on loosely coupled computers, the results are compared, and in case of disagreement a preferred result is identified by majority vote (for  $N = 2$ ) or a predetermined strategy [ELME72, AVIZ77, CHEN78]. The effectiveness of this fault tolerance technique is obviously governed by the degree of independence that can be achieved among the  $N$  versions of the program. It has been recommended that the versions use different algorithms and different programming languages.

A specific constraint on N-version programming is the requirement for  $N$  computers that are loosely coupled yet able to communicate very effectively so that rapid comparisons of the results can be achieved. The SIFT configuration described in 5.5.5 comes close to meeting these requirements. N-version programming is also effective in masking some hardware faults. A disadvantage is that the throughput will depend on the execution time of the slowest of the  $N$  programs that will be executing at any given time.

### 5.6.2 Recovery Block Programming

The recovery block technique can be applied to a wider spectrum of computer configurations, including a single computer (which may include hardware fault tolerance). The key feature of the recovery block is an acceptance test which determines whether the primary routine has furnished an acceptable result. If that is not the case, an alternate is executed [RAND75]. The simplest structure of the recovery block is

Ensure T

By P

Else by Q

Else Error

where T is the acceptance test condition, P represents the primary routine and Q an alternate. Until the results of P or Q are accepted, all data generated by these processes are held in a recovery cache, similar to the cache used with the rollback technique of fault containment.

For flight control and other real-time applications it is necessary that the execution of the program be both correct and on time. To meet the latter condition the acceptance test is augmented by a watchdog timer that monitors the receipt of an acceptable result within a specified period. The structure of a recovery block for programs servicing flight critical functions is shown in Figure 5-12.

In normal operation only the left part of the figure is traversed. When the acceptance test fails or if the time expires a transfer to the alternate call is

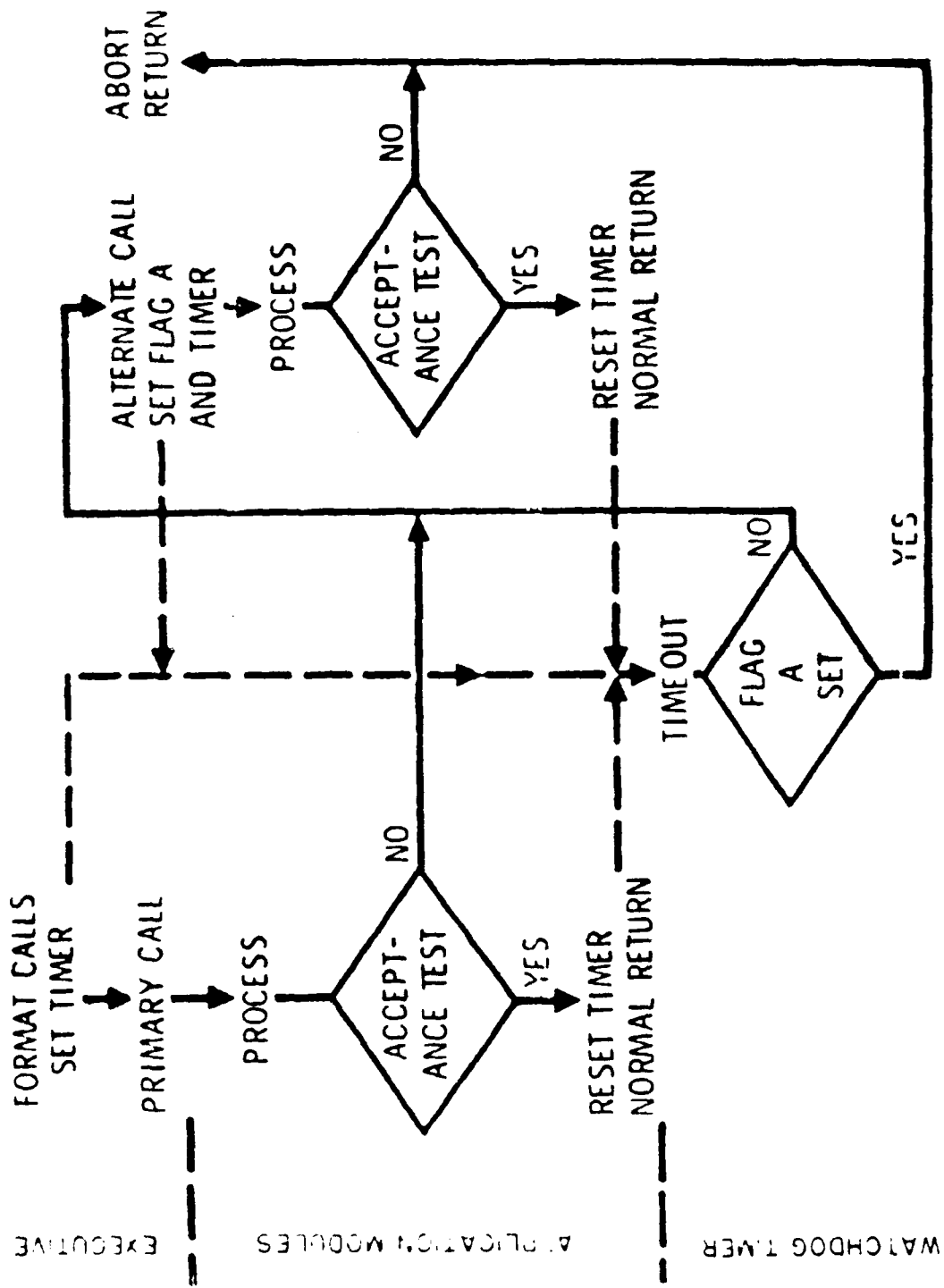


Figure 5 - 12. EXAMPLE OF REAL -TIME RECOVERY BLOCK

initiated, a flag is set, and process Q is executed. If this result satisfies the acceptance test, the normal return exit is taken and processing continues. If the acceptance test fails again or if a timeout is encountered in the execution of Q (with the flag now set), an error return results [HECH79].

The interaction of a real-time recovery block with the system executive is shown in Figure 5-13. The block labeled "Application Modules" and the associated branching for normal and abort returns correspond to the structure shown in Figure 5-12. In fault-free operation, the application modules are called by the Task Select section of the executive, and the normal returns are made back to that section. The same paths are still utilized if a primary application module fails and the alternate succeeds. If none of the application module alternates executes correctly the abort path is entered, and as a first step the failure is recorded. Then a diagnostic program is called which as a minimum determines whether this is a recurrent failure. If the recovery block failure was an isolated instance, the back-up executive may simply suspend the faulty module and cause the Task Select routine of the normal executive to advance to the next application routine. If the failure has been diagnosed as a recurrent one, then a new task schedule has to be generated and substituted for the normal content of Task Select. This may take the form of an Essential Task list which has been decided on in advance, or it may be a modification of the normal task list generated under software control.

The effectiveness of the recovery block technique is largely dependent on the coverage of the acceptance test, i. e., whether that test can detect all significant deviations from normal program execution. For flight critical programs the acceptance criteria can be framed more restrictively: to prevent unsafe output to be furnished to the controlled equipment. The techniques useful for this are largely the same that have been mentioned in connection with fault containment in Section 3. Most flight critical outputs are governed by physical constraints on their range, rate of change, higher derivatives, and frequency response. All of these limitations can be incorporated into the acceptance test.

The acceptance test must be executed every time the recovery block is entered, and this represents the principal throughput penalty associated with this technique. By careful coding of the test, and by using recovery blocks only for the most essential software components, this penalty can usually be kept quite low. The computationally most efficient form of an algorithm can be used for the primary routine. The execution of alternates is so rare that their efficiency is not normally of concern. Because software does not fail in a permanent manner, there are usually automatic switchback provisions to the primary after an alternate has executed a few times.

## 5.7 AIRCRAFT LEVEL FAULT TOLERANCE

The techniques discussed so far are primarily suited to major aircraft systems (flight control, navigation, propulsion) or to subsystems (air data computer, engine controller, etc.). Significant benefits in safety and reliability can be achieved if additional fault tolerance is implemented at the aircraft level. The following techniques involving digital components are particularly pertinent at the aircraft level:

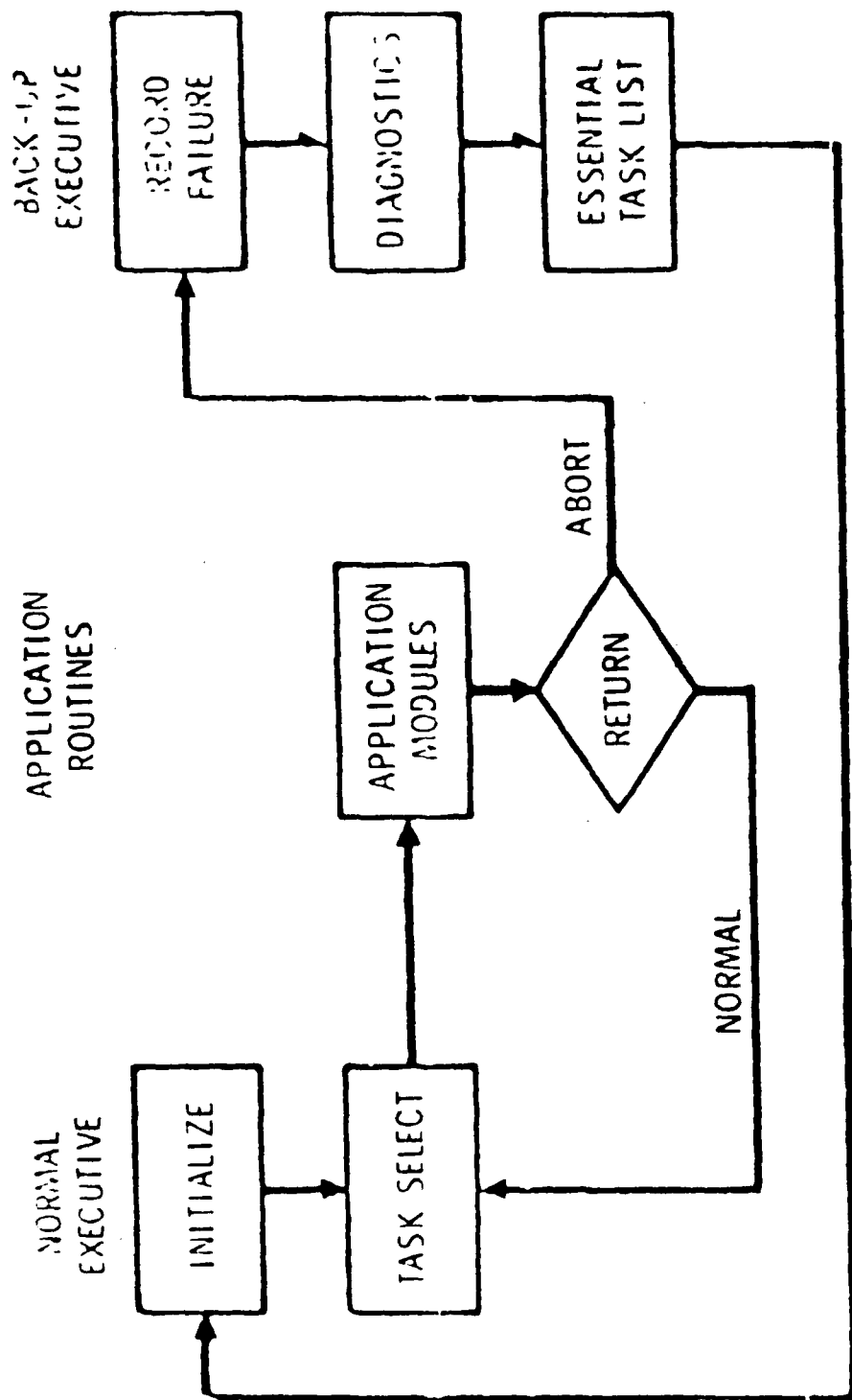


Figure 5 - 13. RECOVERY BLOCK AND SYSTEM EXECUTIVE

- Distributed computing;
- Central malfunction and damage control;
- Use of alternate control modes; and
- Lifeboat systems (independent backup of the most critical functions).

#### 5.7.1 Distributed Computing

In the context of fault tolerance for flight critical systems the essential capability of distributed computing is that a computer from another system or subsystem can either assist in diagnosing a suspected computer or can perform part of the workload of a failed computer. The diagnostic capability can be utilized when there is a disagreement between two computers but the self-test program of neither computer confirms the malfunction. The ability to perform part of the computational tasks of a failed computer is particularly significant when it reduces the need for dedicated redundancy (see 5.8.1).

The reliability of the electronic portions of flight critical systems is generally quite good, e. g., as measured by the absence of these components from the list of most critical aircraft availability and maintenance items. The redundant components are frequently required for safety rather than availability (see Section 5.9). Because failures occur so seldom, some reduced computational performance in case of a failure may be acceptable if it permits elimination of a redundant component (e. g., using a dual system instead of a triplicated one). Distributed computing, properly applied, can permit this.

The provision for standard digital data buses in military aircraft is a very favorable factor for using distributed computing in the manner outlined above. The trend toward standard computer instruction sets will further facilitate the implementation of distributed computing in the context of fault tolerance for flight critical systems.

#### 5.7.2 Central Malfunction and Damage Control

This capability is similar to the one just discussed in that an external computer can assist in fault isolation. It differs from distributed computing in that a central computer is dedicated to fault isolation, reconfiguration, and recovery functions or performs these functions as part of flight systems management. Although this technique is not currently in wide use, it appears economically attractive as the number of flight critical systems that depend on digital components and information increases (the Flight Management Computer in Boeing 757/767 aircraft performs some of these functions). At several instances in this chapter there has been a reference to higher system level fault tolerance provisions. The central computer described here is well suited to this purpose. In many instances the presence of a malfunction and damage

control computer will permit dual systems to achieve the same degree of fault tolerance that otherwise would require a triplicated system.

#### 5.7.3 Use of Alternate Control Modes

In conventional aircraft there is limited capability to use alternate control modes in case of complete failure in a primary control (e. g., use of horizontal stabilizer or trim in case of an elevator malfunction, use of ailerons and rudder trim to compensate for a rudder malfunction). In more recent aircraft designs there are a number of auxiliary surfaces such as leading edge slats, spoilers, and canards which can replace the maneuvering capabilities of primary controls at least over a limited flight envelope.

Integration of the control system to make use of alternate control modes can increase fault tolerance not only with regard to electronic components but also with regard to the non-electronic portions of the primary flight controls.

#### 5.7.4 Lifeboat Systems

A lifeboat is usually less seaworthy than the vessel to which it is attached. Nevertheless, it can save the lives of passengers and crew because its failure modes are largely independent of those of its parent vessel. Similarly, small computers with less functional capability than the primary computers, and without explicit fault tolerance provisions, can perform valuable back-up functions for flight critical systems if their failure modes are independent. Because of the obvious analogy, these minimal back-up computers are referred to as lifeboat systems. A well-known example of a lifeboat system is the single back-up computer in the Space Shuttle Orbiter which has on at least one occasion saved the mission and vehicle when the quadruple redundant primary computers failed.

Failure modes of the primary computer which can be protected against by a minimal independent back-up computer include:

- Timing (either a common clock or a timing adjustment function which can affect multiple computers);
- Hardware design deficiencies;
- Software failures;
- Hardware/software interfaces, particularly in reconfiguration and recovery; and
- Environmental effects (including combat damage).

Except for the last one of these, the failures in the primary system are likely to be of a temporary nature and may not repeat after a restart. Therefore the

lifeboat computer (or the system management function which controls switchover to it) should be capable of re-initializing the primary system and transferring back to it if it is found to be operational.

Switchover to the lifeboat computer can be automatic (particularly when no heartbeat of the primary computer is received), under control of a system management computer, or under crew control. Although the delay associated with the latter mode may not be tolerable under all flight conditions, it may sometimes be the only chance of saving aircraft and crew.

## 5.8 GENERAL APPLICATION NOTES FOR FAULT TOLERANCE

This section summarizes criteria and techniques that apply to many types of fault tolerance and which affect the selection of techniques or their implementation. Specific headings cover the following subjects:

- Partitioning for Fault Tolerance;
- Similar vs. Dissimilar Redundancy;
- Response Time Requirements for Recovery from a Failure; and
- Integration of Fault Tolerance with Diagnostic Capabilities.

### 5.8.1 Partitioning for Fault Tolerance

Some fault tolerance techniques provide or require a specific level of partitioning, e. g., when cyclic error correcting code is applied to memory, the word is the natural partition. Similarly, when a message acknowledgement protocol is used the natural partition of fault tolerance is at the message level. However, most fault tolerance techniques can be applied at a small scope, e. g., at the register level or to bit slices of a CPU, or at a large scope, e. g., applied to an entire computer or an engine control system. The partitioning decisions should consider at least the following:

- Effect on the reliability model;
- Placement of voting or error detection provisions; and
- Effect on system support.

Criteria for selecting a proper scope under each of these headings are briefly described below.



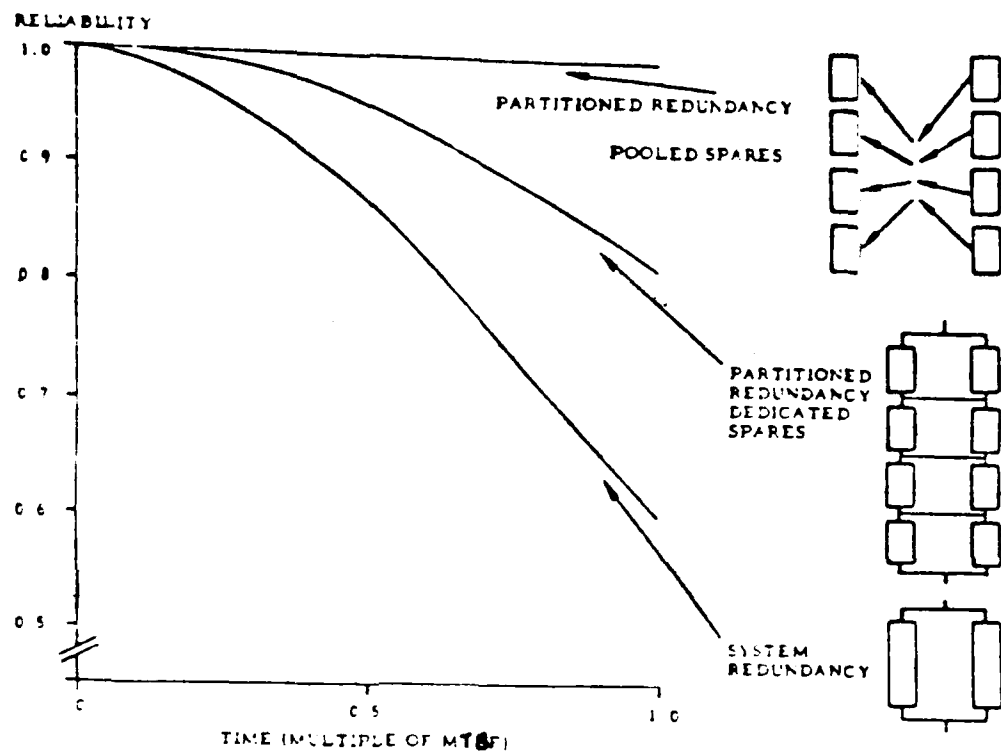


FIGURE 5 - 14 EFFECT OF PARTITIONING ON RELIABILITY

### Partitioning Effects on the Reliability Model

Small partitions for fault tolerance have a significant theoretical advantage because a fault then requires replacement of only a small part of the equipment. This effect is illustrated in Figure 5-14 by comparing the reliability functions for system redundancy and partitioned redundancy - dedicated spares. The single string reliability of each configuration is assumed to be the same (the failure rate of a single system is four times the failure rate of each partition). If the partitioned structure suffers a failure in the top left element and the bottom right element it will remain operational. Where redundancy is applied at the system level the same combination of failures cannot be tolerated. The probability of incurring multiple failures increases with time, and therefore the advantage of partitioning at small scope is seen to be particularly significant if the system has to function without repair for a time approaching one-half of the single string MTBF. Most aircraft missions are of much shorter duration, and if a first failure can be repaired after each mission, these considerations need not affect design decisions as much as where it is expected that many missions must be flown without the possibility for intervening maintenance.

The middle curve in Figure 5-14 provides for a single redundant element in each partition and no possibility for using elements assigned to one partition in another one. This model is applicable at the computer level with one partition representing the CPU, another one memory, and a third one the power supply, etc. Obviously, a spare power supply cannot be used to replace a failed CPU.

The top curve in the figure represents the reliability function for pooled spares, where it is assumed that any spare can replace any failed operational partition. This replacement technique is used in all of the multiprocessor configurations discussed in 5.5.5 and provides a significant reliability advantage over serving the same function with dedicated spares. Note, however, that the magnitude of this advantage is dependent on the time between maintenance actions and that it can be quite small for short mission durations or very reliable single computers. In the figure four pooled spares are shown in order to provide a valid comparison with the other configurations. However, the number of pooled spares need not match the number of active units. Typically, only one or two pooled spares are provided for an arbitrary number of active elements. Pooled sparing is also applicable to redundancy for memory blocks within a computer.

### Placement of Voting and Error Detection

In the preceding heading no allowance had been made for the specific fault tolerance provisions (voting, error detection and reconfiguration, etc.) required for each partition. Factors that need to be considered include:

- Cost, weight and power of the primary circuit elements;
- Resources for self-checking or fault tolerance in the primary elements;
- Increased power and reduced throughput due to external signal routing;
- Time delays introduced by comparison, voting, or decoding; and

- Need to synchronize participating information sources.

In addition, support equipment is affected as indicated in the following heading.

As a result of these considerations, small partitioning is usually not quite as attractive as indicated in Figure 5-14. These factors also suggest that it is highly desirable to place voting and error detection at locations where the information is already in a format that facilitates routing to external elements. Within a computer this is at major interfaces, e. g., from CPU to memory or input/output processors, and within an aircraft system it is at the LRU level, e. g., computer, control panel, or actuator.

#### Effect of Partitioning on System Support

Many system support functions such as documentation, training, test equipment, and spare parts provisioning can be directly affected by partitioning decisions. Partitioning for fault tolerance can implicitly create user (line) replaceable units for which a much greater level of support is required than for units that are replaced only at the depot or by the manufacturer. Because the procurement of manuals and test equipment can involve large costs, the impact of partitioning decisions should be evaluated in the light of the support equipment requirements. Table 5-4 lists a number of support items that may be impacted by partitioning.

TABLE 5 - 4 POTENTIAL SYSTEM SUPPORT IMPACT OF PARTITIONING

Support Function	Required Items
Development	Hardware and Software Specifications
Documentation	Hardware and Software Test Specifications and Reports Software Product Description
User Documentation	Operator Manual User Manual (Function Oriented) Hardware and Software Maintenance Manuals
Test Equipment	Test Equipment Specification (Hardware and Software) Test Equipment Test Specification and Report Test Equipment User Manual Test Equipment Maintenance Manual
Training	Training Requirements/Planning Document Training Facilities Training Program
Configuration	Configuration Management
Control	Change Reports
Spare Parts	Provisioning Documentation Spare Parts Changes to Spare Parts

## Partitioning for Software Fault Tolerance

Because software does not normally fail in a permanent manner, the factors that favor small partitions are largely absent. The scope of fault tolerance provisions for software is usually governed by the availability of good criteria for error detection (particularly in the acceptance test for the recovery block). Where errors in physical quantities are to be monitored, the application of this criterion will usually result in large partitions, e. g., encompassing an entire attitude measurement routine. However, in the system executive functions much smaller partitions may be required because errors need to be detected in quantities such as the number of tasks executed, or the number of available processors. Because each acceptance test impacts the execution time available for the primary functions, very careful design of fault tolerance provisions for the system executive is required.

### 5.8.2 Similar vs. Dissimilar Redundancy

The early efforts in fault tolerant computing were almost exclusively aimed at dealing with random hardware failures, and under these circumstances use of standby or spare units of identical design was not objectionable. Obvious benefits accrue from the use of similar units (meaning of identical design) in a fault tolerant configuration:

- Avoidance of multiple development, including all of the items listed in Table 5-4 above.
- Identical error detection and switching provisions applicable to each unit.
- It facilitates the use of tightly coupled configurations and minimizes the execution time penalty associated with loosely coupled configurations.

On the other hand, software fault tolerance has always had to address design failures and therefore used alternates of different design, even though this involved considerable cost (it negates the benefits just cited). The advent of very large scale integrated semiconductor devices (VLSI) which suffer from many of the same testability limitations as software suggests that some rethinking may be necessary in the choice between similar and dissimilar redundant units for hardware fault tolerance.

At the very least, it must be recognized that fault tolerance that depends exclusively on redundant units of identical design does not protect against failures due to design deficiencies. The lifeboat approach discussed in 5.7.4 offers an effective way of providing fault tolerance based on dissimilar elements. Analytical redundancy of measurements and of derived quantities offers similar advantages. These techniques should be evaluated against a high probability that design related hardware failures will increase in the near future.

### 5.8.3 Response Time Requirements for Recovery from a Failure

As a general rule the complexity of the fault tolerance provisions is inversely related to the time allowed for recovery from a failure. Where the response time has to be extremely short, a voting configuration is usually required and this incurs penalties in physical resources and throughput over simpler alternatives. The tendency to integrate flight control functions (which usually have the shortest response time requirements, of the order of 0.05 seconds) with engine and weapons control may impose more severe recovery limitations on the latter functions and may preclude the use of otherwise acceptable fault tolerance techniques.

To provide an environment in which fault tolerance can be optimized against response time requirements (and in which excessively complex fault tolerance provisions can be avoided) at least the following is suggested:

- Analyze response time requirements for each individual aircraft function to be served by a fault tolerant system.
- Determine maximum fault response time of candidate systems.
- Establish separate fault tolerance partitions for functional requirements that can be served by simpler fault tolerance configurations.

Figure 5-15 represents an example in which the short response time of a voting configuration (the Fault Tolerant Computer in the figure) is utilized to serve time critical flight control functions and also provides a fast response monitoring for external functions (External A and B in the figure). The latter may be associated with weapons or engine control systems. Because the switchover requirements have been assigned to another computer, the units used for A and B do not require any hardware fault tolerance provisions. Their program may need to be modified to provide heartbeat and other diagnostic outputs to the fault tolerant computer.

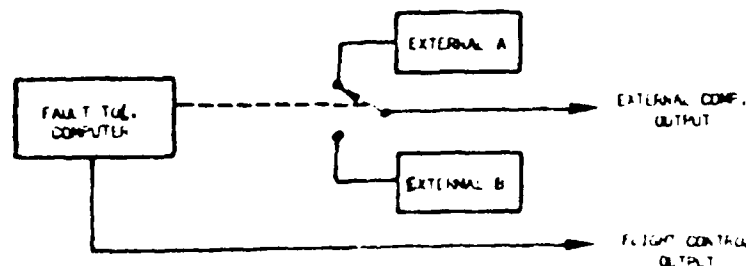


FIGURE 5 - 15 FAST RESPONSE FOR RECONFIGURATION OF EXTERNAL COMPUTERS

#### 5.8.4 Integration of Fault Tolerance and Diagnostic Capabilities

The complexity of digital aircraft equipment makes it necessary to consider testability early in the design, and to incorporate diagnostic capabilities in both the hardware and software components. Guidance for Air Force activities in the testability area is contained in [BYR082]. A military standard for this area, MIL-STD-2165 "Testability Program for Electronic Systems and Equipment" has been issued.

In many cases the requirements for error detection and fault isolation to a line replaceable unit can be met only by providing extensive built-in test (BIT) capabilities. However, where there is also a requirement for fault tolerance, the inherent error detection made available thereby can obviate the need for all or much of the BIT functions. Because considerable resources are required for BIT implementation, integration of these functions with fault tolerance is highly desirable. In practice two situations can arise:

- An LRU is originally designed for applications that do not involve fault tolerance and is later used as part of a fault tolerant system.
- An LRU is intended for fault tolerant applications.

In the former, BIT will be provided for the non-fault tolerant applications and integration can be achieved only by utilizing such BIT capabilities as are found suitable in the fault tolerance provisions. In the latter case, fault tolerance provisions are usually implemented first and then utilized for BIT functions. The reason for the priority assigned to fault tolerance is that error detection and reconfiguration must be carried out under time constraints which are absent in typical BIT scenarios. Thus, trial and error approaches which might be tolerable for BIT purposes must usually be avoided where fault tolerance is required.

Integration of fault tolerance and testability provisions can save both development costs and reduce cost and physical resource requirements for procured equipment. To facilitate this integration the following must be considered:

- Logging of all errors in a medium available to maintenance personnel (non-volatile memory, flight maintenance recorder, etc.).
- Duplication of the on-board fault tolerance provisions in test facilities that may be utilized for further diagnosis.
- Timeliness of error detection in built-in test functions to permit these to be utilized for fault tolerance.

## 5.9 INCORPORATION OF SAFETY OBJECTIVES

In many practical respects the objectives of safety and reliability are identical, and both requirements are served by the fault tolerance techniques discussed in earlier sections of the present chapter. However, some differences must also be recognized: the primary emphasis in reliability activities is on reducing the frequency of failure whereas safety activities concentrate on minimizing the effects of failures. In flight critical systems both need to be accomplished but the following paragraphs deal primarily with safety issues.

The regulations covering safety aspects of flight critical systems have already been described in Chapter 4 of this Handbook. They require the identification of hazardous conditions, and subsequent actions to remove or control these conditions, and to prevent situations which will lead to loss of life, injury, or substantial property damage. A somewhat broader interpretation of safety is implied in MIL-STD-1472 "Human Engineering Design Criteria for Military Systems, Equipment, and Facilities". Paragraph 4.8 of that document states under the heading of Safety :

Consideration shall be given to safety factors, including minimization of potential human error in the operation and maintenance of the system, particularly under the conditions of alert or battle stress.

The automated fault tolerance provisions discussed here fully comply with the intent of that requirement.

Although the effectiveness of fault tolerance provisions described in this chapter may be expressed in terms of reliability, the motivation for fault tolerance arises frequently from safety considerations. This is evident in requiring either a specific maximum failure rate or toleration of a minimum number of failures for a specific function, such as flight control, because that function is recognized to be critical to the safety of the aircraft. In strict reliability terms the improvement effort should focus on systems that have the highest failure rate or contribute the most to mission aborts, regardless of their criticality.

A very significant impact of safety goals on fault tolerance provisions arises in those applications where failure of an output in one direction can produce a much more severe effect on the system than failure in the opposite direction. An example of this type from an advanced aircraft engine control system is shown in Figure 5-16 [MCGL81].

The figure shows the effect of failure of the low pressure turbine inlet control vanes. Four critical engine performance parameters are plotted along the vertical axes, and the power lever angle is plotted along the abscissa. It is seen that failure in the maximum inlet open position will introduce only minor deviations from normal performance whereas failure in the minimum (closed) position will cause much more pronounced deviations.

In particular, the net thrust (FN) is reduced by over 5% when the failure results in closing the inlet control vanes. A reduction in net thrust by more than 2% is usually classified as a criterion for abort. Also, the turbine inlet temperature (T4) can increase by 200 degrees Fahrenheit when the low pressure turbine inlet area is at a minimum. This temperature increase, if it persists

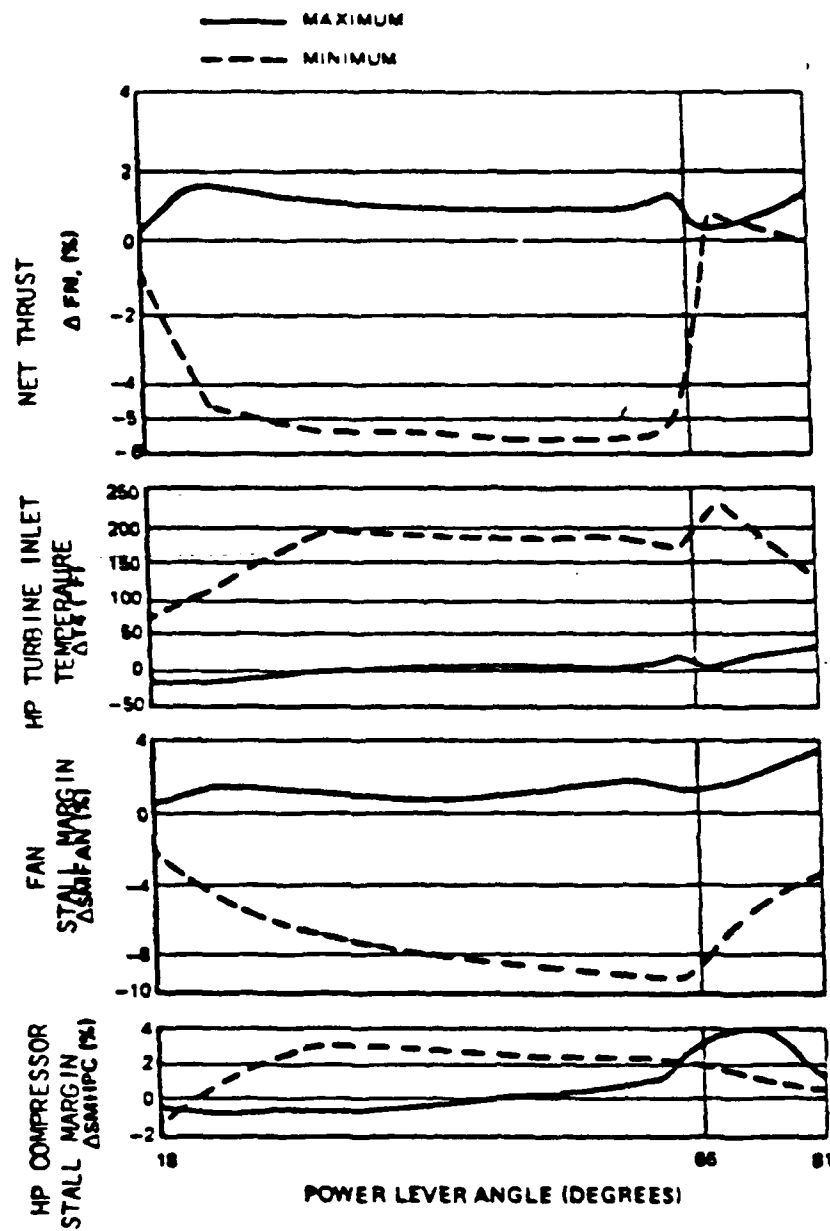


FIGURE 5 - 16 EFFECT OF EXTREME POSITION OF LOW PRESSURE TURBINE INLET CONTROL



for any appreciable time, will cause destruction of the engine. Since the effects of a failure of the inlet control in the open position are much more tolerable than those of a failure in the closed position there is a clear preference for the former case.

A digital computer by itself cannot usually be programmed to provide a safe state after an arbitrary failure. Any such provisions in the software or in the logic can be defeated by an adverse failure of the output elements (which must have the capability of commanding either open or closed for normal control operations). However, electrical, mechanical, or hydraulic provisions can be incorporated at the system level to convert most failures to a safe state. For the low pressure turbine inlet control, an appropriate fail-safe configuration can be achieved by a combination of

- A mechanical or hydraulic pre-load which causes the inlet vane to open in the absence of a control input, and
- Provisions in the engine control computer, in an associated comparator, or in a flight management computer, that cause removal of all input to the inlet vane actuation in case of a detected or suspected failure.

Similar preferred failure modes are encountered in automatic approach and landing systems and in terrain following. In all of these a failure in the fly-up direction is generally preferred to one in the fly-down direction. When fail-operational states have been exhausted, it is therefore desirable to enter a fail-safe mode which implements these preferences.

In other flight critical systems, where a fail-safe mode cannot be defined, some precautions against manifestly unsafe modes can still be incorporated. A minimum requirement is to avoid abrupt changes of signals going to control surfaces. Other areas for implementation of safety after failure are in the transmission of engage/disengage sequences to associated functions. While any one mode of these functions may be tolerated, repeated transitions can be unsafe, particularly if they occur at frequencies that excite aircraft elastic modes.

Any exhaustion of fault tolerance provisions and entry into a "fail-safe" mode must be indicated to the flight crew as a general alarm condition and with identification of the current mode and of any alternative control modes that might be entered by manual selection.

## 5.10 SUMMARY AND TRADE-OFF CRITERIA

A wide spectrum of reliability and fault tolerance techniques has been discussed in this chapter, and the purpose of this concluding section is to summarize these and to provide criteria for their application in flight critical systems. The criteria encompass benefits and costs. The former are expressed in terms of the scope of the reliability improvement and in terms of its effectiveness. The costs are assessed in terms of development risk and in terms of resource requirements for production units. All of the criteria are expressed in Table 5-5 on a net benefit scale of 1 to 5 with 1 representing very low benefits and very high costs, and 5 representing very large benefits and very low costs. Explanation of the individual criteria are presented in the following paragraphs.

### Scope of Improvement

This criterion is based on the fraction of total system level failure modes covered and prevented by the specified technique. The relative probability of the failure modes does not enter into the evaluation because that is highly dependent on the design and quality of each system. In some cases the vast majority of all failures may be due to a computer power supply, and local (power supply) redundancy may greatly reduce the incidence of system level failures. However, power supply failures represent only one of many system level failure modes and the scope of this technique is therefore rated low.

### Effectiveness of Improvement

This criterion is concerned with the completeness of fault avoidance or tolerance within the proper scope of each technique. Thus, reliability improvement techniques have a low score because they cannot be counted on to remove all faults and they provide no fault tolerance. On the other hand, local redundancy is rated high because within its scope it can usually provide high fault coverage. Where the recovery action is expected to result in degraded system performance, the effectiveness is reduced although the fault coverage may be high; this is particularly applicable to system level fault tolerance techniques.

### Development Risk

The development risk criterion evaluates both how much experience exists with a given technique and how consistently it achieves satisfactory results. The latter applies not only to fault avoidance or tolerance, but also to the absence of side effects, particularly in reduction of throughput.

### Resource Requirements

Resource requirements encompass the recurring costs and disadvantages associated with using a technique. Disadvantages include physical resource requirements, loss of throughput, and limitations on interfacing with other subsystems.

TABLE 5 - 5 SYSTEM BENEFIT EVALUATION

Technique	Scope of Improvement	Effectiveness	Development Risk	Resource Requirements
Reliability Improvement	1	2	4	5
Fault Containment	2	3	3	5
Error Correcting Code	2	3	4	5
Local Redundancy	2	4	3	4
Software Fault Tol.	3	4	2	4
Computer Redundancy - Similar Des.	4	4	3	2
Computer Redundancy - Dissimilar Des.	5	5	1	1
System Level Redundancy	5	4	3	2

The scores indicated in the table reflect the results in typical applications and variations by plus or minus one grade may occur in a specific situation. The major objective is to narrow down the choice of techniques to be considered for a stated objective. Thus, if a major reliability improvement is to be attained, the selection should concentrate on techniques that score at least 3 in the scope and effectiveness columns. Similarly, if development risk must be minimized, techniques with a low score in that column should be eliminated at the outset.

## Chapter 6

### EVALUATION METHODOLOGY

Air Force organizations have a major responsibility for the evaluation of aircraft and their components, and this certainly includes the evaluation of flight critical functions and of the equipment associated with these. The present chapter starts with a discussion of evaluation criteria that are pertinent to all phases of a project. This is followed by presentations of specific evaluation methodologies in the order in which they are usually encountered during the life cycle

- Analytic Models
- Simulations
- Evaluation during Development
- Evaluation during Test
- Evaluation during the Operational Phase

The methodologies described in this chapter make extensive use of existing Air Force standards and guidance documents.

#### 6.1 EVALUATION CRITERIA

Evaluation criteria for reliability and fault tolerance requirements form the basis for the evaluation methodology. It is desirable to formulate an outline of the evaluation criteria during the conceptual phase and to add detail as the aircraft and mission become better defined. RAFT evaluation criteria must be completed during the preparation of the Request for Proposal for the developmental model.

##### 6.1.1 Types of Evaluation Criteria

At the aircraft level typical evaluation criteria are

- Abort rate (ground abort, air abort)
- Aircraft losses, fatalities and injuries per flight hour

- Maintenance hours per flight hour
- Unavailability due to maintenance
- Equipment effectiveness (number of functions available/total functions)

At the system and equipment levels quantities derived from the aircraft criteria are more appropriate. Typical evaluation criteria at that level are

- Reliability or failure rate
- Probability of entering an unsafe state
- Availability or downtime ratio
- Number of permanent and transient failures that can be sustained without loss of function
- Number of permanent and transient failures that can be sustained without entering an unsafe state
- Mean time between repair

Because the aircraft (and in many cases also the individual systems) can operate in several modes, a relation between RAFT evaluation criteria and operating modes must be defined. Two approaches are available for this

- The percentage of time that each mode is utilized in a standard mission can be specified, and the criteria are defined as applicable to that standard mission
- Separate criteria are defined for each of the operating modes

The chief advantage of the standard mission approach is the ease of evaluation and simplicity of record keeping. Its chief disadvantages are that it can mask very poor RAFT attributes for modes that have a low weight in the standard mission, and that the relevance of the standard mission to the actual mission requirements is likely to change during the development period. When an evaluation for a new mission profile is desired, this requires contractual negotiations. There is therefore a strong preference for specifying RAFT criteria for each operating mode and to accept the greater effort for evaluation and record keeping that this entails. It will still be desirable to define one or more standard missions so that a simple figure of merit for the RAFT attributes can be generated. However, as the profiles for the standard missions change, new RAFT figures of merit can be generated entirely within the Air Force.

### 6.1.2 Utilization of Criteria

It is highly desirable to document the rationale for the selection of criteria and for the quantitative specifications and to keep the documentation updated as operational requirements change. Considerable expenditures are required for the implementation of reliability and fault tolerance, and it is reasonable to expect that the need for these expenditures will have to be justified repeatedly during the development of an aircraft or a major system. Good documentation of the evaluation criteria can be a major factor in facilitating rational trade-offs of functional and attribute (RAFT) requirements throughout the system life cycle.

In many computer based fault tolerant systems the evaluation criteria for reliability and fault tolerance can utilize the following general classification

- hardware
- software
- fault tolerance provisions
- performance deficiencies

The first two classifications correspond to causes of failure discussed in Chapter 3. Evaluation criteria for the fault tolerance provisions are concerned with the probability of correct error detection, reconfiguration and recovery. The performance deficiency classification deals with failure to meet response time requirements when there is neither a hardware nor a software failure. Performance deficiencies can in some circumstances cause flight critical malfunctions and must be accounted for in applications that require the highest reliability.

RAFT evaluation criteria are sometimes considered as part of contract incentive provisions for flight critical equipment. While this is desirable in principle, it is generally difficult to administer because of

- small lot procurements which introduce a large dispersion into the statistical parameters for reliability and availability
- interaction of functional changes with reliability which will require repeated negotiations of the incentive provisions during the development phase
- uncertain classification of many failures which are likely to lead to disputes regarding the award under the incentive provisions.

The remaining sections of this chapter deal with the evaluation of RAFT attributes during successive life cycle phases.

## 6.2 ANALYTICAL MODELS

Analytical models can be used during concept development and are usually the only tools available during that phase. A general treatment of reliability models for Air Force systems is contained in MIL-STD-756B "Reliability Models and Prediction".

### 6.2.1 Simple Analytical Models

A few formulations for redundant components are described below (these hold for availability as well as for reliability).

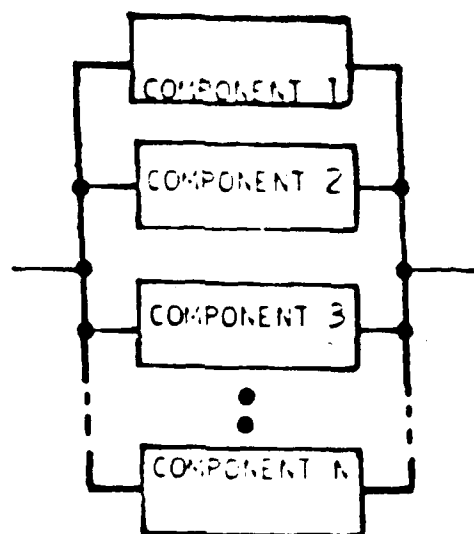


FIGURE 6 - 1 GENERAL FORM OF PARALLEL REDUNDANCY

Redundancy of active elements is illustrated in Figure 6-1. If any one of the parallel components can perform the entire function, then the system failure probability  $F$  is the continued product of all component failure probabilities

$$F_s = \prod_{i=1}^n F_i \quad (1)$$

If all components have the same failure probability the above simplifies to

$$F_s = F_i^n \quad (2)$$

If only two components of equal failure probability are involved the reliability

function becomes

$$R_s = 2 R_f - R_f^2 \quad (3)$$

Configurations which employ pooled spares can be modeled with the binomial or "k out of n" formula. For an aircraft function that requires k components of a given type, and for which n components are initially available, the system reliability is given by

$$R_s = \sum_{i=k}^n \binom{n}{i} R^i (1 - R)^{n-i} \quad (4)$$

This equation assumes that all components are of equal reliability R.

The component failure probability,  $F = 1 - R$ , can be obtained MIL-HDBK-217D "Reliability Prediction for Electronic Equipment". Two methods are described in Section 5 of the Handbook

- the parts stress method, which considers details of the application and of the electrical and thermal stress imposed on the part
- the parts count method, which is based on average application stresses

Prior to the detail design phase there is usually not sufficient information available to use the parts stress method. Once that information is available it becomes the preferred method because parts stress models the failure process more closely than the parts count method. Where a version of the equipment is already in service, or where data on similar equipment is available, these data should be utilized because they include usage factors which otherwise have to be supplied as indicated below. Support of modifications and new developments is an important use of data collection in the operating environment (see also Section 6.6).

#### 6.2.2 Modifications of Simple Models

The simple reliability models have to be modified to account for

- mission phase (e. g., instrument approach equipment enters into the flight control reliability model only during approach)
- environmental conditions (MIL-HDBK-217 lists multipliers applicable to various aircraft types and ground operations)
- software failure probability
- performance deficiencies (see previous section)



As the definition of a system progresses, additional factors must be taken into account, particularly failures due to the fault tolerance provisions. In very simple situations the analytical models described above can be modified to account for imperfect error detection and reconfiguration, but as more refinement is needed a state transition model, such as the one shown in Figure 6-2 will need to be generated.

System level fault tolerance techniques, such as the ones described in Section 5.7, can result in operational states that do not provide full capabilities. The modeling of these degraded states becomes significant for evaluation of the total effectiveness of a new or improved weapon system.

Reliability and availability estimates obtained from analytical models can be important inputs to decisions whether to proceed with a new development. All assumptions, procedures, and intermediate results should be well documented. Analytical models provide more insight into causes of reliability and fault tolerance problems than do the simulation models described below. Therefore analytical modeling is frequently continued even when simulations are available that incorporate a more detailed structure of the error detection and reconfiguration processes.

Analytical and simulation models permit the evaluation of likely outcomes of system or component design decisions. Most of the models predict levels of reliability and availability that can ultimately be achieved if no mistakes are made in the implementation, and thus they tend to be optimistic relative to early operational experience.

### 6.3 SIMULATIONS

Computer simulations of reliability models for fault tolerant systems become necessary as the number of states in the state transition models increases, or as it is desired to include probabilistic inputs. Computer simulations permit much more complex transitions to be modeled, and they also furnish much faster results for simple cases. Figure 6-3 illustrates a state transition model that requires a computer simulation.

Most computer models include probabilistic assumptions for the transition between the states. However, if these transition probabilities change as a result of prior events, or on the basis of flight conditions or computer states two approaches can be taken:

- Each combination of environmental conditions is separately simulated, and the results are combined analytically (or in a higher level computer model), taking into account the probability of encountering the environment of each individual simulation.
- The probability of environmental conditions is input into the computer, and Monte Carlo techniques are used to generate an overall reliability or availability prediction (see MIL-STD-756B Method 1004).

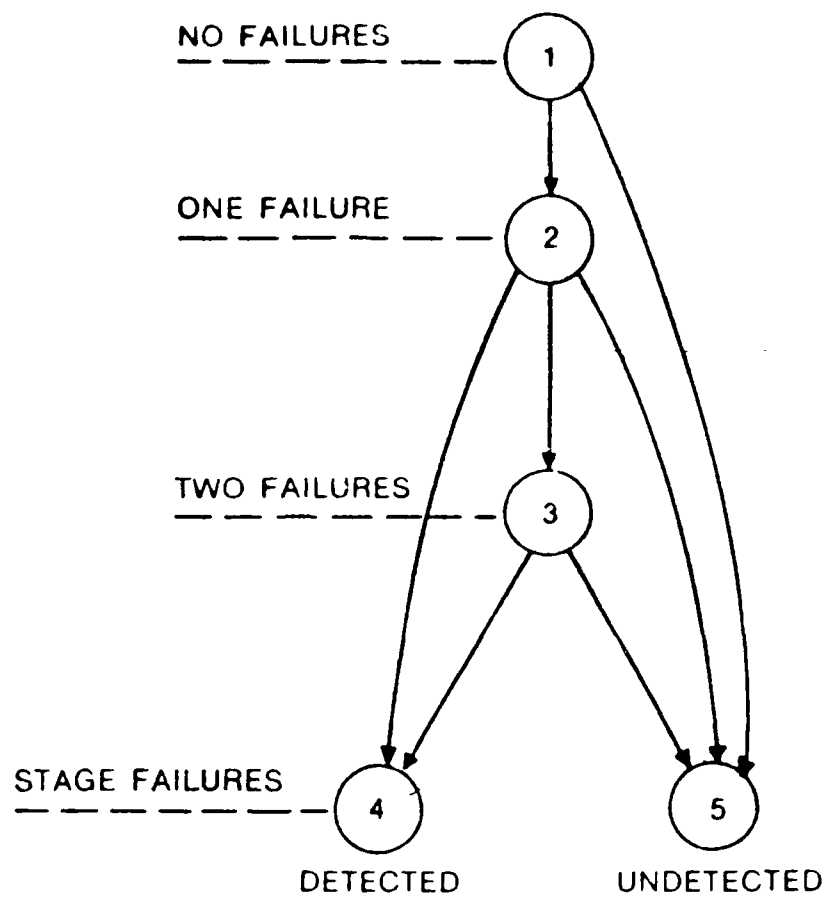


FIGURE 6 - 2 EXAMPLE OF STATE TRANSITION MODEL



The former is called a deterministic simulation, the latter a probabilistic simulation. The deterministic simulation provides better insight but also involves more work. For investigation of multiple environmental conditions the probabilistic approach is usually the only alternative.

In conventional simulations the computer program first simulates the occurrence of a particular failure and then the response to that failure. In terms of Figure 6-3 this means that first one of possibly several hundred failure modes will be simulated that lead to the transition from the OF (no failure) to the 1F (failure of one processor) state, and that then the progression of that particular failure through subsequent system states will be evaluated. The most frequently occurring failures, such as simple memory or processor failures, will also be the most frequently simulated ones. But for the evaluation of flight critical functions the emphasis is on less frequently encountered malfunctions, such as dual failures or latent failures in parts of the recovery provisions, and the small fraction of computer time devoted to these in the conventional simulations makes the process inefficient. A refinement has been introduced in which one simulation (or possibly an analytical model) is used to determine the probability of failure in a specific mode, and the state transition model is used only to evaluate the progression of the failure [GEIS83]. The advantage of this approach is that most of the simulation time can be devoted to the less frequently encountered critical failure modes.

Because the development of a large scale computer simulation is a complex process, a number of general purpose simulations have been developed that can be adapted to the specific configurations that are to be modeled. CARE III is an example of simulations that are well suited to flight critical systems (it has been sponsored by NASA Langley for aircraft applications). CARE is an acronym for Computer Aided Reliability Evaluation. The elementary transition modeled by CARE III is shown in Figure 6-4. The transition probability from state  $i$  ( $S_i$ )

to state  $k$  ( $S_k$ ) is represented by  $r_{ik}$ , and the reverse transition probability is designated by  $r_{ki}$ . Provision for forward and reverse transitions permits

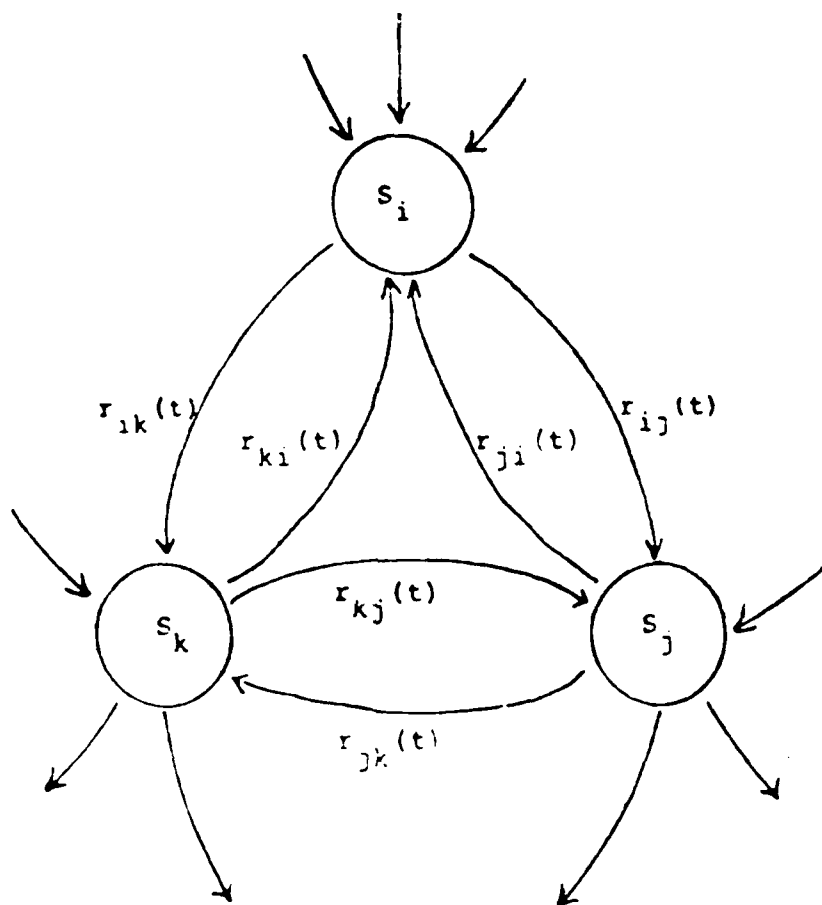
modeling of transient failures. The transition probabilities can be made time dependent, as indicated by the (t) suffix. This permits modeling of delays associated with error detection and reconfiguration.

Documentation of simulations is extremely important. Significant decisions are made on the basis of results obtained from the simulations, and therefore the assumptions, the validation of the simulation, and any changes made after a complete validation need to be described. Because of their cost, simulations frequently carry over from one life cycle stage to another, and may be passed through several organizations. This emphasizes the need for careful documentation.

#### 6.4 RELIABILITY AND FAULT TOLERANCE EVALUATION DURING DEVELOPMENT

A major responsibility in the management of a development program is to identify if and where assumptions of reliability models are being changed, and to either

From Other States



To Other States

FIGURE 1. TYPICAL S. TRANSITION ELEMENTS OF CASE III

reverse the changes or to adjust the models to reflect the new architecture. The availability of well-documented models is of course essential for this task. Closely associated with the general surveillance of the reliability aspects of the system architecture is a continuing investigation of more detailed factors such as

- Performance margins
- Parts derating
- Safety factors applicable to mechanical portions of digital equipment (e. g., mounting brackets)
- Heat generation and removal

Gross failure rate predictions made during earlier phases must be allocated during development to successively smaller partitions of the system. Important considerations are:

- Are the allocated failure rates realistic in view of the actual structure and complexity of the defined components?
- Are failure mode assumptions valid in view of the actual design?
- Can criticality of failures be reduced by modifications of the architecture or by redundancy of small scope?

Any additional information acquired as the development progresses should be used to evaluate the adequacy of fault tolerance and fault isolation provisions. Examples of such information are

- FMECA conducted on portions of flight critical systems
- Fault tree analysis
- Experience with similar systems
- Air Force and National Aviation Safety Board accident reports

One of the most significant documents for the management of the reliability and fault tolerance aspects that is generated during the development phase is the test plan. It should be reviewed with particular emphasis on the following:

- Are all reliability and fault tolerance provisions being tested? In this connection consider the classification of hardware, software, fault tolerance, and performance deficiencies discussed in Section 6.1).
- Is the sequence of tests such that the most critical functions are being tested first (to permit changes in these to be made without requiring extensive retest of less critical functions)?
- Are test cases consistent with the assumptions of the analytical and simulation models?
- Can results of test be compared with those of the models?

- Will results of one series of tests be reported on a schedule that permits review prior to start of the next series?
- Are there provisions for repeating reliability and fault tolerance related tests in case changes affecting these functions are made for other reasons?

## 6.5 RELIABILITY AND FAULT TOLERANCE EVALUATION DURING TEST

Failures experienced during development and pre-test activities can furnish valuable information for the reliability evaluation even if they are not part of the formal test program. The following needs to be investigated for each failure during test:

- Is the failure due to random causes or does it evidence a design deficiency?
- Is the failure mode included in the FMECA? (If not, does this evidence a need for updating the FMECA?).
- Is the frequency of random failures in agreement with the prediction?
- If the failure occurred within a fault tolerant portion of the system, did fault tolerance provisions operate as intended?

It is normal to have an initially high rate of software failures during test. However, this rate (normalized to computer execution time) should decrease with time and at the end of test should be at the specified level. The manifestations of software failures should be reviewed to determine that they are within the capabilities of fault containment or fault tolerance provisions.

If software testing is being conducted on the target computers, the hardware performance during software test can provide important insights into (a) performance bottlenecks which may propagate to reliability problems, and (b) hardware reliability problems.

Stress tests (adverse environments for hardware and high workloads and frequent exception conditions for software) are designed to identify weaknesses in the system and are particularly important for the reliability and fault tolerance evaluation of flight critical systems. Failures that are observed in these tests (including all transient failures) must be investigated thoroughly against the criteria identified earlier in this section.

Failures observed during stress tests (as well as throughout the development and test phases) furnish valuable clues regarding the effectiveness of diagnostic programs. Significant criteria in this respect include:

- Was a pre-failure diagnostic run that might have identified the failure?
- Did a post-failure diagnostic verify the existence of the fault?

- Did post-failure diagnostics identify faults unrelated to the failure, and could these faults be verified in further maintenance actions? False alarms due to diagnostic programs are a perennial problem and seriously detract from their usefulness.
- Is the overall detection frequency of diagnostic programs consistent with the failure frequency observed by other means?

There are a number of generally accepted formats for the reporting of hardware and software tests, and these provide adequate information for the review of reliability and fault tolerance problems. In addition to the overall frequency of hardware and software failures, the following should be investigated:

- Are there any failures which could not be duplicated or verified?
- Are there failures which could be duplicated but which have not been resolved (most likely to arise in connection with software failures)?
- Were there any incidents in which the fault tolerance provisions did not operate exactly as specified (even beneficial deviations from the specification need to be investigated)?
- What are the implications of the test results for the intended operational environment (need to avoid certain flight profiles, restrictions on mode transitions, etc.)?

The test time and the quantity of equipment involved is usually too small to permit evaluation of reliability parameters by classical statistical methods. However, sometimes the experience with these small lots clearly indicates that reliability goals will be difficult to achieve. Contracts should permit such indications to be used to require improvement efforts, alternate development, or termination of the program.

#### 6.6 RELIABILITY AND FAULT TOLERANCE EVALUATION DURING OPERATION

When significant reliability problems are being encountered during the introduction of a new or modified weapon system, ample data are usually available to identify the problem but it may be difficult to find a solution that is economically justified and that does not significantly impact the mission capabilities of the aircraft or system. A sequence of remedial steps may be required, involving

- restricted operation to avoid the specific condition that causes the problem
- inspection and rework where required to eliminate an equipment weakness
- replacement of an equipment by one of a later design that avoids the problem
- redundancy or fault tolerance to overcome the effects of the equipment problem or failures



As each of these steps are being considered or implemented, there is usually a need for reliability evaluations, most likely of the analytic modeling or simulation type. The availability of a current reliability or availability model can be extremely valuable in permitting a timely assessment of the effects of the proposed change.

Another motivation for maintaining analytical reliability models and simulations current during the operational phase is the need to support changes that do not originate in the reliability area. Air Force weapon systems are frequently updated, and to support reliability and fault tolerance activities in connection with such updates the reliability models (both analytical and simulations) must be kept current. This implies periodic reviews of failure rate and failure mode assumptions, maintaining the structure of the model in accordance with the actual aircraft structure (where there are several versions of an aircraft, each will require its own reliability models), and keeping the modelling techniques current with the prevailing methodology.

The evaluation of routine reports is another important reliability activity during the operational phase. Potential data sources include

- Monthly Maintenance Digest, prepared by operational units
- Maintenance Data Collection Record, AFTO Form 349
- Maintenance Discrepancy Report, AFSC Form 258
- Maintenance Actions, Manhours, and Aborts Summary (AFLCR 66-15)
- System, Subsystem and Work Unit Code Failure Summary (prepared from AFTO 349 Forms by AFLC/LOEP)

Most of these reports cover only hardware failures. However, procedures are currently being generated to make AFSC Form 258 applicable to software failures. Software configuration control records sometimes provide an indication of the frequency of software problems.

The above sources are valuable primarily for identifying potential problem areas. To define the complete cause of the problem, its criticality, and possible causes of action will normally require a detailed investigation, using reporting forms or summaries specifically oriented toward a single area.

When there are no catastrophic events, it is frequently very difficult to obtain data to substantiate either the true absence of problems or the existence of problems at a level that does not attract the attention of the operational command but which nevertheless have to be resolved. Typical of the latter are workload related software problems. Since most reliability reports cover a period of one month, and since a much greater fraction of the period represents low workload activity than conditions of high workload, the entries in these reports are rather insensitive to workload related failures. This is an area of concern because the occurrence of failures during a period of high computer workload may have much more serious effects than a similar failure at a low workload levels. High workloads in the flight portions of flight critical systems may be due to

- simulated or actual combat conditions

- failures in non-digital portions of the system
- recovery from failures in the digital system
- turbulence
- control or flight mode transitions

All of these represent conditions under which it is particularly important that the digital system continues to function.

A recommended approach to this potential problem area is to investigate the workload conditions under which failures in flight critical equipment occurred, and to give high priority to the prevention of failures that are associated with high workloads.

## Chapter 7

### APPLICATION OF RELIABILITY AND FAULT TOLERANCE TECHNIQUES

This chapter shows how the techniques described earlier in this Handbook can be implemented in specifications, statements of work, and verification provisions. Three examples dealing with the development and evaluation of flight critical systems in the Air Force environment are presented. The first example deals with the specification of RAFT requirements during the concept phase. The second example is derived from the development phase, and the third represents a reliability improvement program during the operational phase.

In all cases the specific RAFT related requirements discussed here are intended to be used together with functional and performance requirements of the basic equipments, and quality assurance, life cycle cost, and integrated logistics requirements that apply to the weapon system.

#### 7.1 RAFT REQUIREMENTS FOR THE CONCEPT DEFINITION PHASE

The flight critical function discussed in this example is a flight control system which is essential by the definition of MIL-F-9490D only for all-weather operation. It is required that the flight control system be integrated with a weapon delivery system. The latter requires much tighter control accuracy than the 0.5 degree rms error that is adequate for the flight critical function. The integration requirements dictate that a digital flight control system be utilized.

##### 7.1.1 Provisions for the System Specification

The following lists a number of qualitative requirements that are derived from the mission definition presented above that apply to the specification. During the concept definition phase there are usually very few quantitative RAFT requirements, and it is assumed here that there are none. The next subsection covers requirements that affect the conduct of the work and are therefore more suitable for inclusion in a Statement of Work or similar document. The investigations and trade-offs referenced there establish quantitative RAFT requirements during later stages of the life cycle. The rationale is shown indented below each requirement.

1. The performance of the flight control system must not degrade below levels required for all-weather operation following a single permanent part or subsystem malfunction.
  - This is both a common sense requirement arising from the essential nature of the flight control system and a specific interpretation of MIL-F-9490D par. 3.1.3.2.
2. The performance of the flight control system must not degrade below levels required for all-weather operation following two uncorrelated transient errors separated by at least 0.1 seconds. This requirement must be met independent of the occurrence of a permanent part or subsystem malfunction.
  - This requirement reflects the greater probability of transient errors (compared to a permanent malfunction) and the greater capability of coping with them.
  - Independence of the occurrence of a permanent malfunction means that this requirement must be met both before and after the conditions covered by par. 1 above.
3. The responses of the flight control system to the events of the two preceding paragraphs must (a) be deterministic, (b) be recorded in a medium that can be read by maintenance personnel during the next required maintenance, (c) not result in oscillations exceeding the levels of MIL-F-9490D par. 3.1.3.8, (d) be annunciated to the crew if they result in degraded flight control operation or depletion of fault tolerance capability for a period exceeding 1 second.
  - Phrase (a) precludes probabilistic or trial and error approaches to fault isolation; although these are acceptable practices for BIT they are not appropriate as part of the fault tolerance provisions because they may not bring the system to an operational state within the allowable control system delay.
  - Phrase (b) assures that all fault tolerance actions are recorded; this is desirable not only for monitoring purposes but also to preclude the undetected occurrence of transient errors that could interfere with the operation of other fault tolerance or safety related provisions.
  - Phrase (c) avoids significant control transients and also assures that the delay associated with recovery from the error does not cause the aircraft to become unstable. For flight phases other than all-weather landing it is likely that the limits imposed by par. 3.1.3.8 could be safely exceeded.
  - Phrase (d) alerts the crew to degradation of aircraft functions and/or fault tolerance attributes.
4. The system must comply with MIL-F-9490D par. 3.1.3.9 both before and after events of paragraphs 1 and 2 above.

- This requires adherence to minimum quantitative reliability and safety requirements independent of the fault tolerance provisions.
5. Flight readiness shall not be degraded due to the presence of the fault tolerance provisions. Maintenance requirements shall not be increased due to the presence of the fault tolerance provisions.
- These requirements are intended to preclude configurations which will violate them very grossly; more precise requirements will be formulated as part of the studies undertaken during the concept definition phase.
6. Single malfunctions in each of the interfacing systems and utilities shall not cause an error, transient or permanent, in the flight critical functions.
- This represents a minimum requirement for control of the interfaces; it also implies that normal outputs or performance of these systems must not cause transient or permanent errors in the flight critical functions.

This list is deliberately free of references to specific fault tolerance configurations, reliability practices, or interfacing requirements that might constrain the choice of an approach.

#### 7.1.2 Studies and Activities Requirements

The requirements stated below apply to the concept definition phase and may be met by Air Force in-house activities (including services of a support contractor) or they may be included in a Statement of Work for a contractor. A common rationale for all activities is to provide a basis for specific quantitative and qualitative requirements during later phases.

##### 1. Conduct the following activities in accordance with MIL-STD-785B "Reliability Program for Systems and Equipment"

###### 1.1. Task 101 -- Reliability Program Plan covering all activities indented below:

- This establishes a framework for controlling the activities.

###### 1.2. Task 103 -- Program Reviews to be conducted at specified intervals and covering tasks to be completed at those times.

- The conventional sequence of MIL-STD-1521 reviews is usually not applicable to the definition phase; hence specific designation of review dates and objectives is required.

###### 1.3. Task 201 -- Reliability Modeling in accordance with Tasks 101 and 102 of MIL-STD-756B (any suitable method at the option of the implementer) and including consideration of hardware, software, and fault tolerance failures.

- This includes the modeling and simulation activities discussed in Chapter 6 of this Handbook.

1.4. Task 203 — Reliability Prediction in accordance with Tasks 201 and 202 of MIL-STD-756B (any suitable method at the option of the implementer) and including consideration of hardware, software, and fault tolerance failures.

- This task provides the parameters for the reliability modeling defined in the previous task.

1.5. Task 204 — Failure Modes, Effects, and Criticality Analysis at the system and subsystem levels.

- Identifies critical failure modes and interfaces.

1.6. Task 208 -- Reliability Critical Items as derived from Task 204

- This establishes identification and control of flight critical items and interfaces.

2. Conduct the following activities in accordance with MIL-STD-470A "Maintainability Program Requirements".

2.1. Task 101 -- Maintainability Program Plan

- Establishes the framework for all other intended activities.

2.2. Task 206 — Maintainability Design Criteria Plan with emphasis on validation of fault tolerance provisions after maintenance.

- The basic document establishes criteria for diagnosability, access to equipment, interchangeability of parts; it needs to be tailored for fault tolerant equipment.

2.3. Task 201 — Maintainability Modeling with emphasis on maintenance of the fault tolerance provisions.

- This document forms the basis for assessment of downtime, readiness, and total maintenance resource requirements.

2.4. Task 203 — Maintainability Prediction

- Provides the parameters for the maintainability model.

3. Subsystem Design Analysis Reports in accordance with DI-S-3581 on the following subjects:

3.1. Single Point Failure Avoidance

3.2. Fault Tolerance/Operational Readiness Trade Study

3.3. Fault Tolerance/Maintainability

- These reports provide qualitative and quantitative data for system design decisions in later phases.

### 7.1.3 Verification Provisions

A state transition simulation is the key element of the verification of RAFT attributes during the concept definition phase. The simulation shall preferably be developed and operated independent of the system development in order to avoid misunderstandings of the specification from affecting both the system under development and the simulation. The simulation can be used to verify:

- Single permanent fault tolerance,
- Double transient fault tolerance, and
- Speed of recovery (the simulation must be instrumented to generate the delay associated with each recovery step and to output the sum of the delays).

The other specification provisions are verified by the reports generated under the Studies and Activities requirements.

## 7.2 RAFT REQUIREMENTS FOR THE DEVELOPMENT PHASE

The flight critical system in this example is the automatic control of a jet engine. A manual backup control system is available that is adequate for subsonic flight but not for the supersonic regime. In terms of MIL-F-9490D the automatic control is classified as flight phase essential. Fault tolerance is required only to permit safe transition to the subsonic regime in which the pilot can assume manual control. Quantitative requirements cited in this example are for illustrative purposes only and should not be interpreted as being typical of current needs or capabilities. In practice the quantitative requirements will be developed as part of the trade studies during the concept definition or validation phases. A preliminary specification containing these values may be circulated to potential bidders in order to solicit comments on their ability to comply.

### 7.2.1 Provisions for the System Specification

In contrast to the avoidance of structural detail in the concept definition phase specification (see preceding section), the development phase specification assumes a configuration that has been identified as the optimum implementation in preceding studies.

1. The automatic engine control system shall consist of a primary channel that provides full control during all phases of flight and a secondary channel that permits safe deceleration from supersonic to subsonic speeds if the primary channel fails during supersonic flight.

- The dissimilar design approach was found practical because the secondary channel can be kept very simple. Thus, protection against design failures can be provided without incurring the cost penalties usually associated with dissimilar designs.

2. Operation of the primary and secondary channels shall be completely independent between the power lever shaft and the mechanical output of the engine actuators.

- This implies independent sensors as well as separate electric and hydraulic supplies.

3. Switching from the primary channel to the secondary channel shall occur under the following conditions:

- primary channel self-check fails,
- turbine inlet temperature exceeds high limit, and
- pilot selects secondary.

4. All transitions from primary to secondary shall be annunciated by the engine warning system. Transition from secondary to primary shall be possible only when initiated by the pilot. Operation of the secondary channel shall be monitored by the engine warning system. Both channels shall complete a self-check at least once per second. All self-check failures and all transitions shall be transmitted to the digital flight recorder.

- Self-checks rather than cross-checks are used to preserve the independence of the systems. The engine warning system, though primarily intended to monitor mechanical and thermal problems, acts as a further check and enables the pilot to switch between primary and secondary. Malfunctions of the engine control can be handled within the limitations of human reaction time.

5. The failure rate of the primary automatic engine control system shall not exceed 0.003 per flight hour, 0.001 per non-flight power-on hour, and 0.0003 per hour when not powered. Pertinent failures include those due to hardware, software, and performance causes but do not include those due to faulty maintenance, exposure to environments outside the specification limits or induced by accidents or failures in other aircraft systems. Failure rates for the secondary channel shall not exceed one-third of the values listed above.

- The reliability and maintainability requirements of the primary channel were derived (allocated) from an aircraft availability specification. The requirements for the secondary channel were made equivalent (based on functional complexity) to those of the primary channel.

6. The primary channel shall not require scheduled maintenance more often than once per 100 flight hours and 300 non-flight power-on hours. The secondary channel shall not require scheduled maintenance more often than once per 300 flight hours and 1,000 non-flight hours. When not powered neither channel shall require scheduled maintenance more often than once every 3 years.

7. Scheduled maintenance of either channel shall not require more than 30 minutes on an aircraft in which standard access is provided. Ninety percent of all unscheduled maintenance actions shall not require more than 30 minutes and no unscheduled maintenance action shall require more than 3 hours.



8. Switchover and annunciation provisions shall be checked as part of routine post-flight inspections.

#### 7.2.2 Studies and Activities Requirements

The requirements include all those described under this heading in Section 1. Since development is assumed to be contracted out the provisions will in all cases be incorporated in the statement of work. The scope of the requirements has to be adapted to the development environment, e. g., in MIL-STD-785B Task 103 the program reviews specified in the governing contract schedule are invoked. In addition, the following will be invoked:

##### 1. Reliability Program in accordance with MIL-STD-785B

###### 1.1. Task 102 -- Monitor/Control of Subcontractors and Suppliers

- To be invoked on both hardware and software suppliers.

###### 1.2. Task 104 -- Failure Analysis, Reporting, and Corrective Action (FRACAS) with reporting to begin as shown below

Component	Test Phase for FRACAS Reporting
Sample and qualification lots	All tests
Parts Subassemblies	Acceptance test
Line replaceable units and system	Functional test
Software	Following unit test
Rejected and repaired items	All tests

FRACAS Summary Reports shall be furnished monthly.

1.3. Task 105 -- Failure Review Board. A single administrative body shall be responsible for both hardware and software failures. Detail review activities may be conducted as delegated. The Air Force Reliability Engineer (or similar position) shall be a member of the Failure Review Board.

- The major responsibility is the timely compliance with FRACAS requirements.

###### 1.4. Task 205 -- Sneak Circuit Analysis for all engage, disengage, and annunciation functions;

- routine analysis and test of these functions does not usually provide complete coverage.

1.5. Task 201 -- Parts Program

- A routine requirement in support of the reliability requirements

2. MIL-STD-470A Maintainability Program

2.1. All tasks listed under this heading in 7.1.2

2.2. Task 202 -- Maintainability Allocation

2.3. Maintainability Demonstration Plan in accordance with MIL-STD-471A and DI-R-5318.

- Both of these activities are required in support of the maintainability objectives of the specification.
- After Review of the Maintainability Demonstration Plan a decision shall be made whether it is economically feasible to conduct a demonstration in accordance with MIL-STD-470A Task 301.

7.2.3 Verification Provisions

Analysis and state transition simulations (as described in the preceding section) continue to be key RAFT verification provisions during the early stages of the development phase. As hardware becomes available, they will be augmented by:

- Component breadboards and demonstration models.
- System bench, usually connected to a simulator for the flight environment. The former permit verification of functional requirements (e. g., that the secondary channel provides safe deceleration to the subsonic region), and the system bench permits verification of the switchover provisions. At still later stages the system bench will be populated with production equipment which permits realistic evaluation of RAFT related performance and timing characteristics.

The number of components procured during a development program is seldom sufficient to conduct a formal reliability demonstration in accordance with MIL-STD-781C. Instead the following provisions may be incorporated in the contract or referenced documents:

1. Compliance with hardware reliability requirements shall be demonstrated by analysis. However, the requirements shall be deemed not to have been met if hardware failures experienced during a designated test period (which should include the software tests to take advantage of the operating time accumulated there) show at a confidence level of 75% or greater that the failure rate exceeds the specified value.
- This will reject the system if the failure rate is significantly higher than specified; it does not provide assurance that the failure rate is within specification.

2. Software reliability shall be measured in terms of failure rate averaged over one month. The failure rate is a fraction, the numerator of which is the number of observed failures and the denominator of which is the CPU utilization in hours during the reporting interval. The trend of the failure rate, obtained by linear regression over a six month interval, shall be negative until a failure rate of 0.05 per CPU-hour has been achieved. There shall be no failures in flight critical portions of the software during the last 320 operating hours prior to acceptance.
3. Software development and software quality assurance methodologies shall be specified to be compliant with the two applicable standards, MIL-STD-2167 and MIL-STD-2168.
  - These standards incorporate requirements for in-process reviews that provide good evaluation of software quality.
4. It shall be required that flight critical software be subjected to 100% (or near-100%) complete branch and call testing (see Section 5.6).
  - This is a minimum requirement. Branch and call testing does not provide total test coverage since details of the execution may depend on data values or the computer state. The near-100% option may be taken when there are calls or branches that can be accessed only when certain interface conditions are true which cannot be attained in the test environment.
5. Subsystem Design Analysis Reports in accordance with DI-S-3581 on the following subjects:
  1. Single Point Failure Avoidance
  2. Fault Tolerance/Operational Readiness Trade Study
  3. Fault Tolerance/Maintainability
  - During the development phase these reports are appropriate for evaluation (to determine that attributes assumed in the design decisions were not materially changed in the implementation).

### 7.3 REQUIREMENTS FOR A RELIABILITY IMPROVEMENT PROGRAM

In an advanced trainer the stall warning is currently furnished through the air data system which employs a digital computer. The computer has a high failure rate which has not been significantly reduced through several modifications that had been recommended by the manufacturer. Because a failure of the computer will simultaneously disable the primary air data display and the stall warning, trainees have frequently had to rely entirely on the secondary airspeed and altitude indicators and several near accidents have been reported under these conditions. Three remedial measures have been considered:

- making the air data system redundant,
- a complete redesign of the air data computer, and
- providing a stall warning system that is independent of the air data computer.

The first of these has been rejected because it will add an unacceptable amount of weight and will result in a non-standard cockpit configuration. The second alternative is considered to be high risk because previous modifications have not achieved the desired reliability improvement. A separate stall warning system is available from several vendors. The objective of the aircraft reliability improvement program is to procure a stall warning system that will improve the in flight safety (by reducing dependence on the air data computer), will not significantly add to the aircraft maintenance requirements, and will not detract from the dispatch readiness. The latter two requirements can only be met by a system that has high inherent reliability (reliability without recourse to redundancy or fault containment).

#### 7.3.1 Provisions for the System Specification

The system specification emphasizes independence from the existing equipment and high reliability.

1. The operation of the stall warning system shall be completely independent of the existing air data system. It shall be capable of operating from the utility electric power bus.

- The air data system operates from an EMI protected bus. The small power requirements of the stall warning system make it feasible to provide EMI protection as part of the equipment, and thus independence of electric power supplies can be achieved.

2. The stall warning system shall contain self-check provisions covering the electric input to electric output portion. Self-check failure shall result in an instrument warning. The operation of the self-check shall be capable of being monitored by maintenance personnel without recourse to external test equipment.

- These provisions are essential because operation of the system is not routinely observable by the pilot.

3. The failure rate of the stall warning system shall not exceed 0.001 per flight hour, 0.0001 per non-flight power-on hour, and 0.00003 per hour when not powered. Pertinent failures include those due to hardware, software, and performance causes but do not include those due to faulty maintenance, exposure to environments outside the specification limits or induced by accidents or failures in other aircraft systems.

- These reliability requirements are demanding but are necessary to avoid impacting the maintainability and readiness of the aircraft.

4. Ninety percent of all maintenance actions shall be completed in less than 1 hour.

- Because there will be comparatively few maintenance actions, a tight control of maintenance time is not considered essential. Higher requirements will add to development and evaluation costs without compensating operational benefits.

#### 7.3.2 Studies and Activities Requirements

Only reliability and maintainability provisions applicable to production equipment are involved. There is no requirement for trade studies. Since a large number of equipments will be procured and since a low failure rate is an important requirement, a formal reliability demonstration program is provided for.

##### 1. Reliability Program in accordance with MIL-STD-785B

1.1. Task 101 -- Reliability Program Plan

1.2. Task 102 -- Monitor/Control of Subcontractors and Suppliers

1.3. Task 104 -- Failure Reporting, Analysis and Corrective Action System for acceptance testing of parts and subassemblies and system testing of all higher components

1.4. Task 105 -- Failure Review Board for hardware and software

1.5. Task 203 -- Reliability Prediction covering both hardware and software, using MIL-STD-756A Method 2001 (Similar Item Method)

- Since this equipment is already in production, the similar item method will provide the most realistic prediction.

1.6. Task 303 -- Reliability Qualification Test Program in accordance with MIL-STD-781 Test Plan IIIC. Test to be run under simulated flight conditions.

- The expected decision point for this test is reached at approximately 11 times the specified MTBF (in this case 1,000 hours). If 10 systems are used for the test, it can be completed in approximately 45 days.

2. Maintainability Program in accordance with MIL-STD-1689 including

2.1. Task 202 — Maintainability Allocation

2.2. Task 301 -- Maintainability Demonstration

- Both of these activities are required in support of the maintainability objectives of the specification.

### 7.3.3 Verification Provisions

The RAFT related verification provisions are satisfied by the reliability demonstration and maintainability demonstration included in the previous subsection.

# REFERENCE:

- AVIZ77 A. Avizienis and Liming Chen, "On the Implementation of N-Version Programming for Software Fault Tolerance During Program Execution", Proceedings Compsac 77 (Computer Software & Applications Conference), Chicago, pp. 149-155, November 1977
- AVIZ82 Algirdas Avizienis, "The Four-Universe Information System Model for the Study of Fault Tolerance", Digest of Papers 12th Fault Tolerant Computing Symposium, IEEE Cat 82CH1760-8, pp. 1157-1191
- BAKE82 L.E. Baker et al, "Full-Authority Fault-Tolerant Electronic Engine Control System for Variable Cycle Engines". Report AFWAL-TR-82-2037, June 1982.
- BERL80 Elwyn R. Berlekamp, "The Technology of Error-Correcting Codes", Proceedings of the IEEE, pp. 564, May 1980
- BOSS81 D. C. Bossen and M. Y. Hsiao, "ED/FI: A Technique For Improving Computer System RAS", Digest, FTCS-11, IEEE Cat No. 81CH1600-6, pp. 2-6, June 1981
- BYR82 J. Byron, L. Deight and G. Stratton, "RAIC Testability Notebook", RAIC-TR-82-189, June 1982
- CHEN78 L. Chen and A. Avizienis, "N-Version Programming: A Fault Tolerance Approach to Reliability of Software Operation", Digest of Papers, FTCS-8, pp 3-9, June 1978
- CLIF70 D.P. Clifford, "All-Weather Automatic Landing Development and Testing", Annals of Reliability and Maintainability, 1970, pp. 75-81, SAE, New York, July 1970
- ELME72 W.R. Elmendorf, "Fault Tolerant Programming", Digest of Papers, 1972 International Symposium on Fault Tolerant Computing, IEEE Catalog No. 72CH0623-9C, June 1972
- FIPS83 FIPS PUB 99 "Guideline: A Framework for the Evaluation and Comparison of Software Development Tools", National Bureau of Standards, March 1983
- FREU45 Alfred M. Freudenthal, "The Safety of Structures", Transactions of the Am. Soc. of Civil Engineers, October 1945, pp. 1157-1191
- GEI83 R.M. Geist, K.S. Trivedi, J.B. Dugan and M.K. Smotherman, "Design of the Hybrid Automated Reliability Predictor", Proc. 5th IEEE Digital Avionics Systems Conference, November 1983
- HECH79 H. Hecht, "Fault Tolerant Software", IEEE Trans. on Reliability, vol R-28 no 3 pp 227-232, August 1979

- HECH81 H. Hecht, "Final Technical Report, Reliability Support of the Fault Tolerant Spaceborne Computer (FTCS)", SoHAF Incorporated TR-81-1, Prepared for USAF Space Division under contract F04701-78-C-0087, February 1981.
- HECH82 H. Hecht, "VLSI - Benefits and Problems for Fault Tolerant Design", GOMAC 1982 Digest of Papers, pp. 192-195, November 1982
- HECH82A H. Hecht and M. Hecht, "Use of Fault Trees for the Design of Recovery Blocks", Digest of Papers, FTCS-12, pp 134-139, IEEE Cat No. 82CH1760-8, June 1982
- HECH83B H. Hecht and M. Hecht, "Trends in Software Reliability for Digital Flight Control", NASA Ames Research Center, April 1983
- HOPK78 A. L. Hopkins et al., "A Highly Reliable Fault Tolerant Multiprocessor for Aircraft", Proc. of the IEEE, vol 66, no 10, pp. 1221-1239, October 1978
- HOUG81 R.C. Houghton, Editor, "Proceedings of the NBS/UEEE/ACM Software Tool Fair", NBS Special Publication 500-80, October 1981
- HOUG82 R.C. Houghton, "Software Development Tools", NBS Special Publication 500-88, March 1982
- HOWL78 W.E. Howden, "An Evaluation of the Effectiveness of Symbolic Testing", Software-Practice and Experience, vol 8, pp. 381-397, John Wiley & Sons, 1978
- IEEE83 Standard Glossary of Software Engineering, IEEE Std. 729-1983
- KECE64 D. Kececioglu and D. Cormier, "Designing a Specified Reliability Directly into a Component", Proc. of the Third Annual Aerospace Reliability and Maintainability Conference, pp. 546-565, SAE, July 1964
- LARI81 S.J. Larimer and S.L. Maher, "A Continuous Reconfiguring Multi-Microprocessor Flight Control System", Wright Patterson Air Force Base AFWL-TR-81-3070, May 1981
- LOSQ75 J. Losq, "Influence of Fault-Detection and Switching Mechanisms on the Reliability of Standby Systems", Digest of the 1975 International Symposium on Fault Tolerant Computing, IEEE Cat 75CH0974-6C, pp. 81-86
- LUSS57 Robert Lusser, "Predicting Reliability", Research and Development Division, Ordnance Missile Laboratories, Redstone Arsenal, Alabama, October 1957
- MACK83a D.A. Mackall, V.A. Regenie and M. Gordo, "Qualification of the AFTI/F-16 Digital Flight Control System", NAECON Paper 324, May 1983
- MACK83b D.A. Mackall, "AFTI/F-16 Digital Flight Control System Experience", First Annual NASA Aircraft Controls Workshop, Langley Research Center, October 1983



- MCCO80 S. F. McConnell and D. F. Stewioren, "C-MU Voter Chip", Report CMU-CS-80-107, Carnegie-Mellon University, March 1980
- MCIN83 J.W. McIntee, "Fault Tree Technique as Applied to Software", Rev. March 1983. Available from the Author at BMC/AWS, Norton AFB, Ca 92409
- MCGL81 M.E. McGlone et al., "Full Authority Fault-Tolerant Electronic Engine Control System for Variable Cycle Engines", AFWAL Aero-Propulsion Laboratory, AFWAL-TR-81-2121, July 1981
- MCEL85 Robert J. McEliece, "The Reliability of Computer Memories", Scientific American, pp 88, January 1985
- NASA79 NASA Langley Research Center, "Validation Methods Research for Fault Tolerant Avionics and Control Systems, Working Group Meeting II", NASA Conference Publication 2130, October 1979
- OLMA82 M.D. Olman, "Quantitative Fault Tree Analysis Using the Set Evaluation Program (SEP)", NUREG/CR-1935, September 1982
- RAND75 B. Randell, "System Structure for Software Fault Tolerance", IEEE Trans. on Software Engineering, vol SE-1 no 2 pp. 220-232, June 1975
- RANG79 E.R. Rang et al., "Digital Flight Control Software Validation Study", AFFDL-TR-79-3076, June 1979
- RAO74 T. R. N. Rao, Error Coding for Arithmetic Processors, Academic Press, New York, 1974
- REDI84 H.A. Rediess and E.C. Buckley, "Technology Review of Flight Critical Flight Control Systems", NASA CR172332, NASA Langley Research Center, April 1984
- ROSS82 D. J. Rossetti and R. K. Iyer, "Software Related Failures in the IBM 3081: A Relationship with System Utilization", Stanford University, Center for Reliable Computing, CRC Technical Report 82-8, July 1982
- RUBE75 Raymond J. Rubey, "Quantitative Aspects of Software Validation", Proceedings of the 1975 International Conference on Reliable Software, IEEE Cat No. 75CH0940-7CSR, pp. 246-251, April 1982
- SAIB82 S. H. Saib, "Automated Verification of Flight Software - User's Manual", NASA Contractor Report 166346, April 1982
- SMIT83 T.B. Smith and J.H. Lala, "Development and Evaluation of a Fault Tolerant Multiprocessor (FTMP)", NASA R166071, May 1983
- SMIT84 T.B. Smith, "Fault Tolerant Processor Concepts and Operation", Digest of Papers, The Fourteenth International Conference on Fault-Tolerant Computing, pp. 158, June 1984

- SEAL76 K.J. Szalai, C.R. Jarvis et al., "Digital Fly-by-Wire Flight Control Validation Experience", NASA TM72660, December 1978
- SZAL77 K.J. Szalai, R.R. Larsen and R.D. Glover, "Flight Experience with Flight Control Redundancy Management" in Fault Tolerance Design and Redundancy Management Techniques, AGARD Lecture Series No. 109, NATO, Neuilly sur Seine, 1960
- TAYL80 D.J. Taylor et al., "Redundancy in Data Structures: Improving Software Fault Tolerance", IEEE Trans. Software Eng., vol SE-6, no. 6, pp. 585-595, November 1980
- TAYL81 J.R. Taylor, "Fault Tree and Cause Consequence Analysis for Computer Software Validation", RISO (Denmark) National Laboratory, Report M-2326, 1981
- VESE80 W.E. Vesely et al., "Frantic II - A Computer Code for Time Dependent Unavailability Analysis", NUREG/CR-1924, July 1980
- WENS78 J. H. Wensley et al., "SIFT: The Design and Analysis of a Fault Tolerant Computer for Aircraft Control", Proc. of the IEEE, vol 66 no 10, pp. 1240-1254, October 1978

## Appendix A

### GLOSSARY

This appendix consists of two parts. The first is a list of abbreviations and acronyms used in the body of the report, and the second part gives the complete nomenclature for DoD documents, primarily standards and specifications, that are referred to only by alphanumeric symbols in the text.

#### A.1 ABBREVIATIONS AND ACRONYMS

AFALC	USAF Air Logistics Command
AFTI	Advanced Fighter Technology Integration (Program)
AFWAL	Air Force Wright Aeronautical Laboratories
AIPS	(NASA) Advanced Information Processing System
ARINC	Aeronautical Radio Incorporated
ASD	(USAF) Aeronautical Systems Division
AVFS	Automatic Verification of Flight Software
BIT	Built-In Test
CPU	Central Processing Unit
CRMMP	Continuously Reconfiguring Multi-Microprocessor
DFCS	Digital Flight Control System
DoD	Department of Defense
EMI	Electromagnetic Interference
FAA	Federal Aviation Authority
FCS	Flight Control System
FMEA/FMECA	Failure Modes, Effect (and Criticality) Analysis
FTMP	Fault Tolerant Multi-Processor

HCL	High Order Language
IEEE	Institute of Electrical and Electronic Engineers
IFFN	Identification Friend, Foe, Neutral
IL	Integrated Logistics
LRU	Line Replaceable Unit
MTBF	Mean Time Between Failures
NASA	National Aeronautics and Space Administration
PDR	Preliminary Design Review
RPV	Remotely Piloted Vehicle
SIFT	Software Implemented Fault Tolerance
TMR	Triple Modular Redundancy
VDC	Volt Direct Current
VHSIC	Very High Speed Integrated Circuits
VLSI	Very Large Scale Integration
XOR	Exclusive "or"

#### A.2 FULL NOMENCLATURE OF GOVERNMENT DOCUMENTS

DOD-STD-2167	Military Standard, "Defense System Software Development"
DOD-STD-2168	Military Standard, "Software Quality Evaluation"
MIL-F-9490	Military Specification, "Flight Control Systems - Design, Installation and Test of Piloted Aircraft"
MIL-HDBK-217	Military Handbook "Reliability Prediction of Electronic Equipment"
MIL-HDBK-472	Military Handbook "Maintainability Prediction"
MIL-Q-9858	Military Specification, "Quality Program Requirements"
MIL-STD-470	Military Standard, "Maintainability Program for Systems and Equipment"
MIL-STD-471	Military Standard, "Maintainability Verification/ Demonstration/ Evaluation"
MIL-STD-490	Military Standard, "Specification Practices"

MIL-STD-704	Military Standard, "Aircraft Electric Power Characteristics"
MIL-STD-796	Military Standard, "Reliability Modeling and Prediction"
MIL-STD-781	Military Standard, "Reliability Design Qualification and Acceptance Tests: Exponential Distribution"
MIL-STD-785	Military Standard, "Reliability Program for Systems and Equipment Development and Production"
MIL-STD-882	Military Standard, "System Safety Program Requirements"
MIL-STD-883	Military Standard, "Test Methods and Procedures for Microelectronics"
MIL-STD-1472	Military Standard, "Human Engineering Design Criteria for Military Systems, Equipment and Facilities"
MIL-STD-1521	Military Standard, "Technical Reviews and Audits for Systems, Equipment and Computer Programs"
MIL-STD-1553	Military Standard, "Aircraft Internal Time Division Command/Response Multiplex Data Bus"
MIL-STD-1629	Military Standard, "Procedures for Performing a Failure Mode, Effects and Criticality Analysis"

## Appendix B

### EXCERPTS FROM FEDERAL AVIATION REGULATIONS

This appendix contains excerpts from a number of Federal Aviation Regulations (FARs) which are used by the Federal Aviation Administration to certify civil aircraft. Military aircraft are not obligated to adhere to these requirements. The excerpts are presented because it is believed that familiarity with them will aid in establishing suitable criteria for flight critical systems in general.

The FARs have evolved as a result of experience in the use of designs, equipment, and maintenance practices, and their present structure does not correspond to a systematic analysis of aircraft functions. They are presented below in numerical order:

1. FAR 25:671           Control Systems
2. FAR 25:672           Stability Augmentation and Automatic and Power-Operated Systems
3. FAR:25:673          Two-Controlled Airplanes
4. FAR 25:1309         Equipment, Systems, and Installation
5. FAR 25:1329         Automatic Pilot System
6. FAR 25:1333         Instruments Systems
7. FAR 25:1351         Electrical Systems and Equipment

#### 1. Control Systems (25.671)

(c) The Airplane must be shown by analysis, test, or both to be capable of continued safe flight and landing after any of the following failures...

- (1) Any single failure, excluding jamming.
- (2) Any combinations of failure not shown to be extremely improbable, excluding jamming (for example, dual electrical/hydraulic system failures, or any single failure in combination with any probable hydraulic or electrical failure).
- (3) Any jam in a control position normally encountered during takeoff, climb, cruise, normal turns, descent and landing unless the jam is shown to be extremely improbable, or can be alleviated. A runaway of a flight control to an adverse position and jam must be accounted for if such runaway and subsequent jamming is not extremely improbable.

- (d) The Airplane must be designed so that it is controllable if all engines fail. Compliance with this requirement may be shown by analysis where that method has been shown to be reliable.

2. Stability Augmentation and Automatic and Power-Operated Systems (25.672)

If the functioning of stability augmentation or other automatic or power-operated systems is necessary to show compliance with the flight characteristics requirements of this part, such systems must comply with paragraph 25.671 and the following:

- (b) The design of the stability augmentation system or of any other automatic or power-operated system must permit initial counteraction of failures of the types specified in paragraph 25.671(c) without requiring exceptional pilot skill or strength, by either the deactivation of the system, or a failed portion thereof, or by overriding the failure by movement of the flight controls in the normal sense.
- (c) It must be shown that after any single failure of the stability augmentation system or any other automatic or power-operating system:
  - (1) The airplane is safely controllable when the failure or malfunction occurs at any speed or altitude within the approved operating limitations that is critical for the type of failure being considered.
  - (2) The controllability and maneuverability requirements of this part are met within a practical operational flight envelope . . . .
  - (3) The trim, stability, and stall characteristics are not impaired below a level needed to permit continued safe flight and landing.

3. Two-Controlled Airplanes (25.673)

Two-Controlled airplanes must be able to continue safely in flight and landing if any one connecting element in the directional-lateral flight control system fails.

4. Equipment Systems and Installation (25.1309)

- (b) The airplane system and associated components, considered separately and in relation to other systems, must be designed so that:
  - (1) The occurrence of any failure condition which would prevent the continued safe flight and landing of the airplane is extremely improbable.
  - (2) The occurrence of any other failure conditions which would result in injury to the occupants or reduce the capability of the airplane or the crew to cope with adverse operating conditions is improbable.

(d) Compliance with the requirements of paragraphs (b) and (c) of this section must be shown by analysis, and where necessary, by appropriate ground, flight, or flight simulated tests. The analysis must consider

- (1) Possible modes of failure, including malfunctions and damage from external sources,
- (2) The probability of multiple failures and undetected failures,
- (3) The resulting effects on the airplane and occupants, considering the stage of flight and operating conditions, and
- (4) Crew warning cues, corrective action required, and the capability of detecting faults.

(e) Each installation whose functioning is required by this subchapter, and that requires a power supply, is an "essential load" on the power supply. The power sources and the system must be able to supply the following power loads in probable operating combinations and for probable duration:

- (1) Loads connected to the system with the system functioning normally.
- (2) Essential loads, after failure of any one prime mover, power converter, or energy storage device.
- (3) Essential loads after failure of:
  - (i) Any one engine on two- or three-engine airplanes, and
  - (ii) Any two engines on four- or more-engine airplanes.
- (4) Essential loads for which an alternate source of power is required by this chapter, after any failure or malfunction in any one power supply system, distribution system or other utilization system.

#### 5. Automatic Pilot System (25.1329)

- (a) Each automatic pilot system must be approved and must be designed so that the automatic pilot can be quickly and positively disengaged by the pilots to prevent it from interfering with their control of the airplane.
- (f) The system must be designed and adjusted so that, within the range of adjustment available to the human pilot, it cannot produce hazardous loads on the airplane, or create hazardous deviations in the flightpath, under any condition of flight appropriate to its use, either during normal operation, or in the event of a malfunction, assuming that corrective action begins within a reasonable period of time.
- (g) If the automatic flight integrates signals from auxiliary controls or furnishes signals for operation of other equipment, there must be positive interlocks and sequencing of engagement to prevent improper operation. Protection against adverse interaction of integrated components, resulting from a malfunction, is also required.



6. Instrument Systems (25.1333)

For systems that operate the instruments required by paragraph 25.1303(b), which are located at each pilot's station:

- (b) The equipment, systems, and installation must be designed so that one display of the information essential to the safety of flight which is provided by the instruments, including attitude, direction, airspeed, and altitude will remain available to the pilots, without additional crew member action, after any single failure or combination of failures that is not shown to be extremely improbable; and
- (c) Additional instruments, systems, or equipment may not be connected to the operating system for the required instruments, unless provisions are made to insure the continued normal functioning of the required instruments in the event of any malfunction of the additional instruments, systems, or equipment which is not shown to be extremely improbable.

7. Electrical Systems and Equipment

- (b) Generating system. The generating system includes electrical power sources, main power busses, transmission cables, associated control, regulation, and protective devices. It must be designed so that:
  - (1) Power sources function properly when independent and when connected in combination;
  - (2) No failure or malfunction of any power source can create a hazard or impair the ability of remaining sources to supply essential loads.

## Appendix C

### BIBLIOGRAPHY

This Appendix lists a number of significant publications which are not found in the Reference section of the report. The list is divided into three sections:

1. General (addressing broad topics in the employment of flight critical digital systems),
2. Specific Applications (either design implementations or studies of specific applications), and
3. Specific Techniques (such as reliability, software engineering, etc.).

Within each section the documents are listed in inverse chronological order (most recent first).

#### C.1 GENERAL

E. F. Hitt, J. Webb, C. Lucius, M. S. Bridgman (Battelle Columbus Laboratories) and D. Eldredge (FAA Technical Center), "Handbook - Volume I, Validation of Digital Systems in Avionics and Flight Control Applications", FAA Technical Center, Atlantic City, DOT/FAA/CT-82/115, July 1983

W. G. Ness, R. M. Davis, J. W. Benson, M. K. Smith (Lockheed-Georgia Co.) and D. Eldredge (FAA Technical Center), "Integrated Assurance Assessment of a Reconfigurable Digital Flight Control System", FAA Technical Center, Atlantic City, DOT/FAA/CT-82/154, April 1983

J. G. McGough, Kurt Moses (Bendix Corp.) and J. F. Klafin (Grumman Aerospace Corp.), "Advanced Flight Control System Study", NASA Ames Research Center, CR-163120, November 1982

D. B. Mulcare, W. C. Ness, and R. M. Davis (Lockheed-Georgia Co.) "Digital Flight Control System Validation Technology Assessment", NASA Ames Research Center, CR-166374 (also DOT/FAA/CT-82/140), July 1982

D. B. Mulcare, W. G. Ness, J. M. McCarty, J. M. Richards (Lockheed-Georgia Co.), E. O. Throndsen, W. J. Hillman (Lockheed-California Co.), D. L. Hemmel, E. P. Kosowski (Rockwell International, Collins Avionics Div.), and E. F. Hitt (Battelle Columbus Laboratories), "Industry Perspective on Simulation Methods and Research for Validation and Failure Effects Analysis of Advanced Digital Flight Control/Avionics", NASA Ames Research Center, CR-152234, February 1979

## C.2 SPECIFIC APPLICATIONS

AFWAL Flight Dynamics Laboratory (Lt. D. Russ), "Terrain Following/Terrain Avoidance Algorithm Study", AFWAL TR-85-3007, In preparation 1985

AFWAL Flight Dynamics Laboratory (R. Bortner), "Multi-Micro Processor Flight Control System II", AFWAL TR-84-3076, In preparation 1985

Northrop Corp., "Proposed MIL-PRIME Formatted Flight Control Specification", AFWAL/FIGL TR-84-3114, December 1984

University of Colorado, Multivariable Flight Control Design with Uncertain Parameters", AFWAL/FIGL TR-83-3036, September 1983

AFWAL Flight Dynamics Laboratory (Lt. Russ), "Proceedings of the Workshop on Multi-Variable Control Systems", AFWAL/FIGL TR-83-3098, September 1983

HR Textron, Inc., "Study of Alternate Approaches for Digitally Controlled Flight Control Actuators", AFWAL/FIGL TR-83-3041, July 1983

S. Osder (Sperry Flight Systems), "DC-9-80 Digital Flight Guidance System Monitoring Techniques", AIAA Journal of Guidance and Control, Jan-Feb 1981

C. Rabinowitz, R. Otterberg, K. Boucher, K. Walworth, P. Cote, J. Vernon (Hamilton Standard Div., United Technologies Corp.) and M. McGlone (Pratt and Whitney Div., United Technologies Corp.), "Reliability Advancement for Electronic Engine Controllers", AFWAL-TR-80-2063, August 1980

## C.3 SPECIFIC TECHNIQUES

IEEE Computer Society, Computer (Magazine), Special Issue on Fault Tolerant Computing, Vol. 17 No. 8, August 1984

S. S. Osder (Sperry Flight Systems), "Generic Faults and Design Solutions for Flight-Critical Systems", AIAA Guidance and Controls Conference, San Diego CA, August 1982

D. P. Siewiorek (Carnegie Mellon Univ.) and R. S. Swarz (Prime Computer Inc.), The Theory and Practice of Reliable System Design, Digital Press, Billerica MA, 1982

P. DeFeo (NASA Ames Res. Ctr.) and S. Saib (General Research Corp.), "A Digital Flight Control System Verification Laboratory", National Aerospace and Electronics Conf. (NAECON), Dayton OH, May 1982

B. W. Boehm (TRW), Software Engineering Economics, Prentice Hall, Englewood Cliffs NJ, 1981

by E. L. Lyle and R. L. Lyle, *Reproductive Management, Methods, and Medications*,  
Second Edition, By the Authors, Redondo Beach CA, 1977



## Appendix D

### EXPERIENCE WITH FLIGHT CRITICAL DIGITAL SYSTEMS

Information relating to flight critical digital systems, primarily flight control systems and components, is presented under two headings: production installations and experimental installations. Only very limited data are available at the current time, and the approach taken here is to report incidents rather than to attempt a statistical evaluation.

#### D.1 PRODUCTION INSTALLATIONS

There are no flight critical digital systems in current service with the U. S. Air Force but the F-16 Digital Flight Control System is in full-scale development as this report is being prepared. An overall survey of production digital flight control systems is shown in Table D-1. Most aircraft on which these are installed have backup provisions, and in those applications the systems are considered non-flight critical. The only production equipment in use for critical applications is the flight control system for the Space Shuttle Orbiter. The other flight critical applications are expected to undergo first flights during the next few years. Most of the data in the table were obtained from [REDI84].

None of the systems in Table D-1 is routinely utilized by the Air Force, and maintenance records for these were not available for analysis. Instead, failures in other digital components were examined for possible impact on flight critical functions. Data for this analysis were obtained from the records of the Air Force Inspection and Safety Center for the following aircraft: F-15, F-16, F-111, and C-5A. These incidents are summarized in Table D-2. There were no injuries, fatalities or major aircraft damage due to these mishaps.

TABLE D - 1 PRODUCTION DIGITAL FLIGHT CONTROL INSTALLATIONS

Aircraft	Year First Flight	No. of Channels	Remarks
NON-FLIGHT CRITICAL INSTALLATIONS			
JA37 Viggen (Sweden)	1974	1	Mechanical backup
DC-10 Autopilot	1975		Not in all aircraft
Tornado Autopilot (Europe)	1976	2	Digital outer loop, mechanical and analog inner loop
CH53E DFCS	1977	2	Mechanical backup, marginal handling qualities
F-18 DFCS	1978	4	Mechanical backup in pitch, analog in roll/yaw
MD-80 DFCS	1981	2	Mechanical primary controls
Boeing 767/757	1982	3	Mechanical primary controls analog backup spoilers
A-310 Spoilers (Europe)	1982	2	Dissimilar hardware and software
Agusta A-129 Helicopter (Italy)	1983	2	Mechanical back-up except tail rotor
FLIGHT CRITICAL APPLICATIONS			
Space Shuttle	1979	4 + 1	Single channel runs dissimilar software
JAS-39 (Sweden)	1985	3	No backup
Lavi (Israel)	1986	1	Analog backup
JVX Tilt Rotor	1987	3	Optical fiber transmission
F-16 DFCS		4	

TABLE D - 2 USAF AIRCRAFT INCIDENTS INVOLVING DIGITAL SYSTEMS

Aircraft	System	No. of Reports	Remarks
F-15	Communications	1	Short circuit in RFI filter caused smoldering
F-15	Inertial Nav.	2	Nav. failures caused problems only when aux. attitude display was not working
F-16	Inertial Nav.	13	Most failures involved frozen attitude displ. sometimes without warning. Some CVD.
F-111	Weapons Computer	1	Shorted connector permitted gun to fire with firing port closed
C-5A	Inertial Nav.	0	

Except for the F-16 Navigation System failures, these data do not indicate any characteristics of digital systems that set them apart from other aircraft electronic equipment. Many of the F-16 failures occurred during company service readiness or service acceptance flights and may have been due to inadequate checkout of the digital components. Nevertheless, these incidents raise concerns that:

- serious in-flight failures could not be duplicated on post-flight maintenance, and
- failures occurred without warning indicators (flags) showing that anything was wrong with the equipment.

In at least one case the post-flight diagnostics suggest that the failure may have been caused by software.

The decision to install a digital flight control system on the F-16 represents a significant milestone in the acceptance of digital technology for flight critical functions within USAF. The major components involved in this system and its interfaces are shown in Figure D-1. A block diagram of the digital computer is shown in Figure D-2. Of the four independent computer channels three furnish direct control outputs to servo valves (SVx) while the fourth channel acts primarily as a monitor as shown in Figure D-3. This arrangement holds true for the five flight control surfaces that are controlled by the integrated servo actuators (ISAs): left and right horizontal tail, rudder, and left and right flaps. The leading edge flaps are controlled by dual motors, and to furnish outputs to these the four channels are arranged in a dual-dual configuration as shown in Figure D-4.

## D.2 EXPERIMENTAL INSTALLATIONS

Significant experience with the use of digital systems for flight critical functions has been reported from joint USAF/NASA projects at Edwards Air Force Base/Dryden Flight Research Facility. The first of these projects was the F-8 Digital Fly-by-Wire (DFBW) control system which used a surplus Apollo guidance computer. The mechanical controls were removed from the aircraft but an analog surface control system was retained as a back-up. This phase of the project accumulated 58 hours of flight time in 42 flights with no report of significant reliability problems [SZAL78].

The second phase of that project involved a triplex digital flight control installation using computers of more recent design. The programming and check-out of the triplex installation exposed a number of problems in computer information interchange and synchronization. The first flight of this configuration occurred in 1976 and the project continued until 1980 at which time over 80 flights were conducted, totaling approximately 100 hours. The incidents reported from that phase are summarized in Table D-3 [SZAL80].



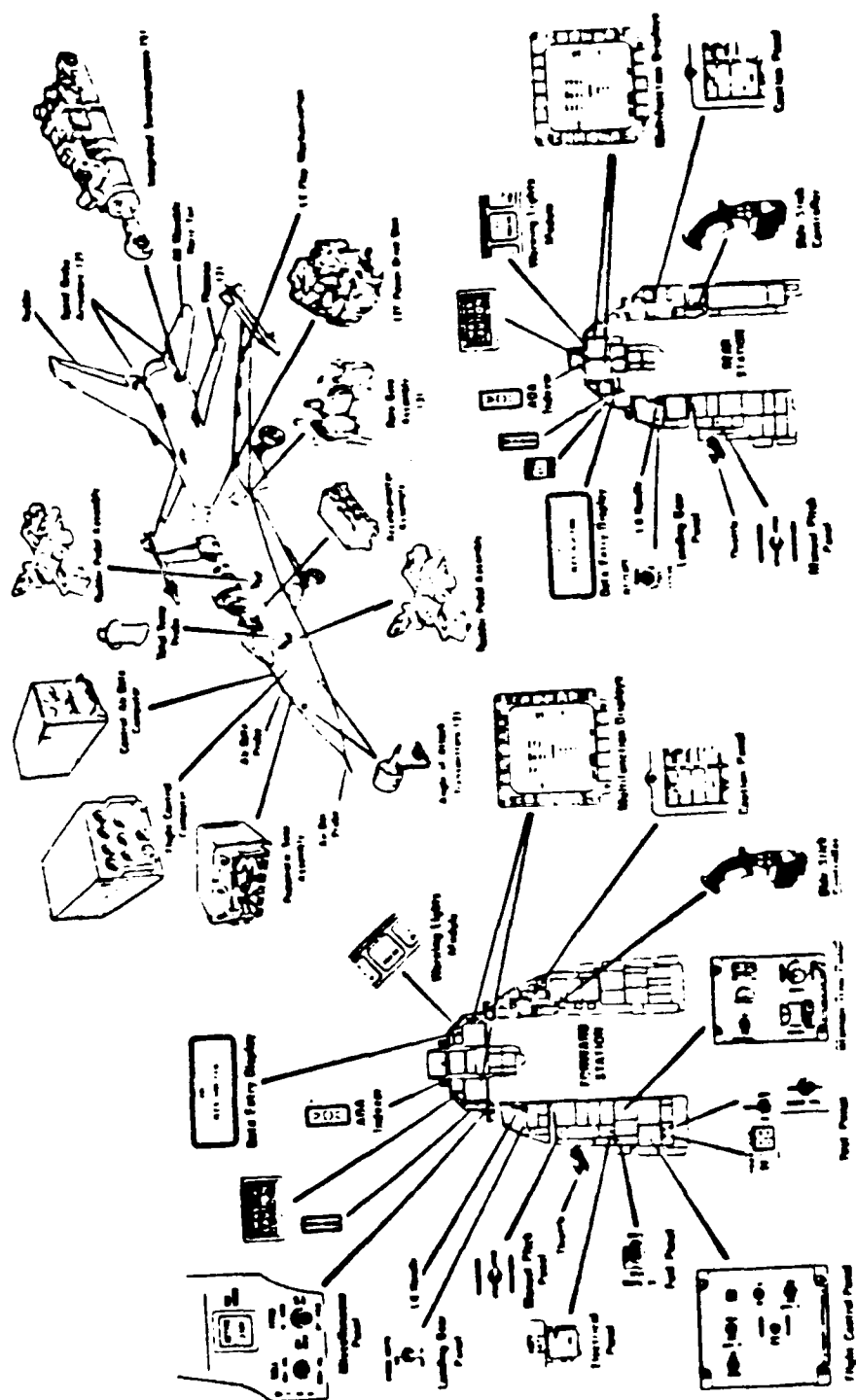


FIGURE D-1 F-16 DIGITAL FLIGHT CONTROL SYSTEM

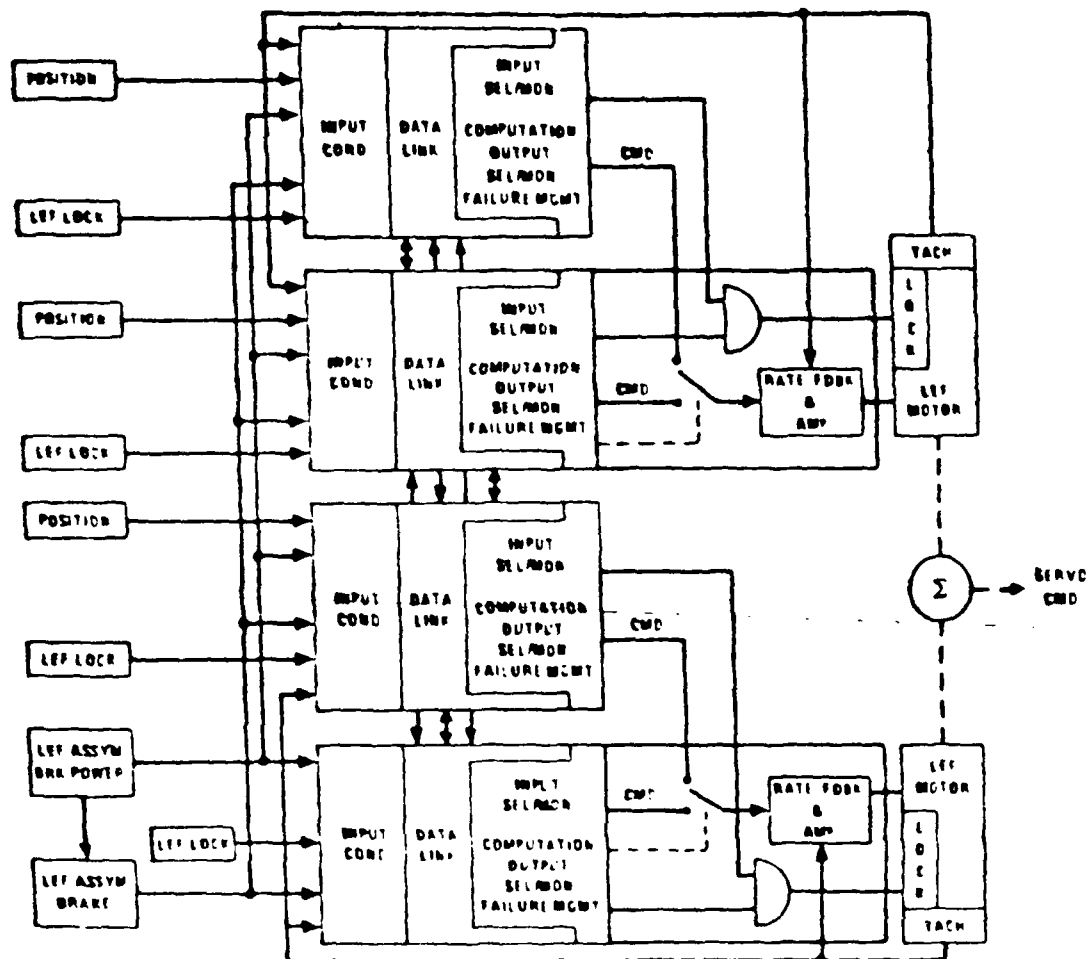


FIGURE D-2 F-16 FLIGHT CONTROL COMPUTER

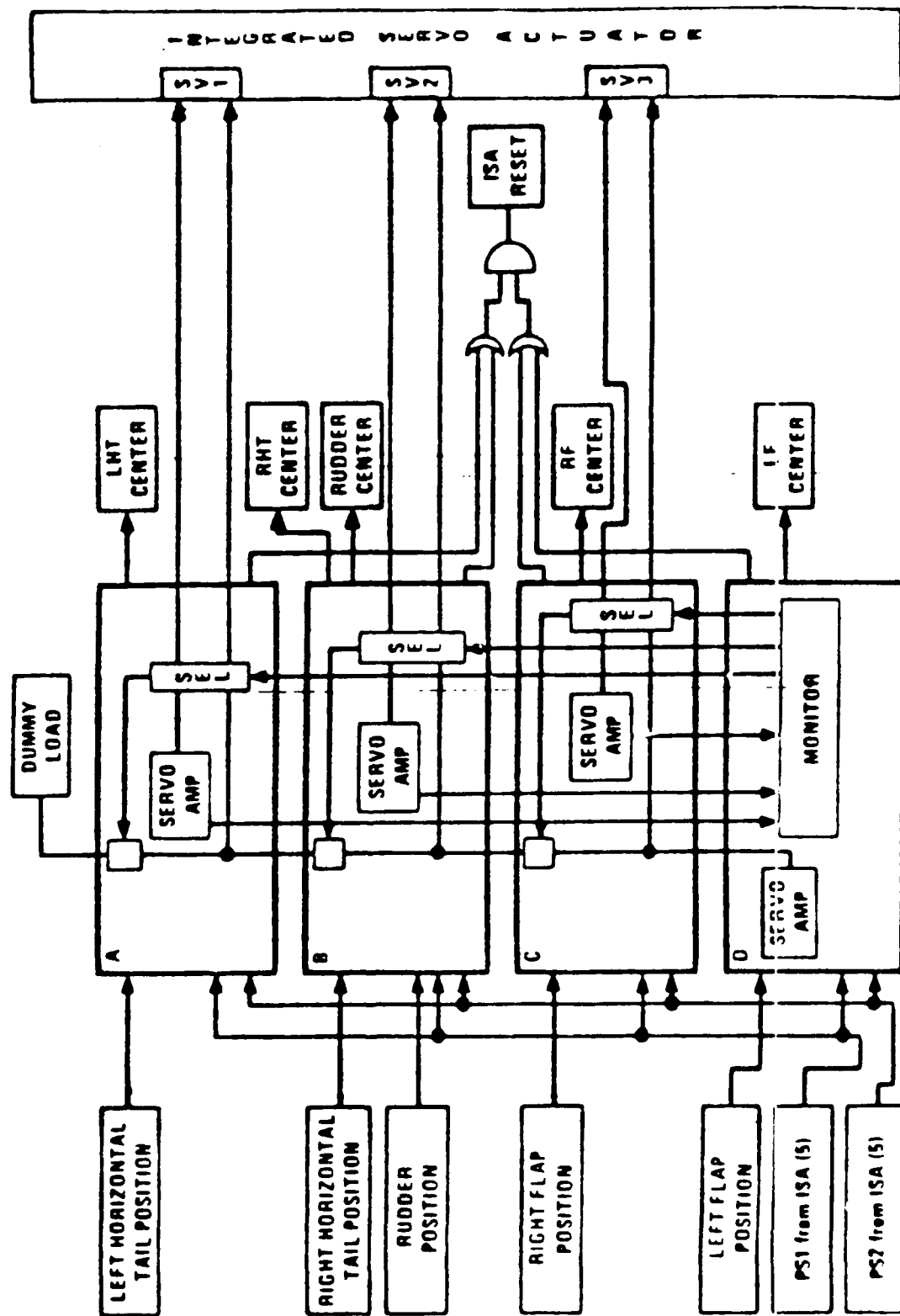


FIGURE D-2 SERVO ACTUATOR INTERFACES OF THE F-16 DECS

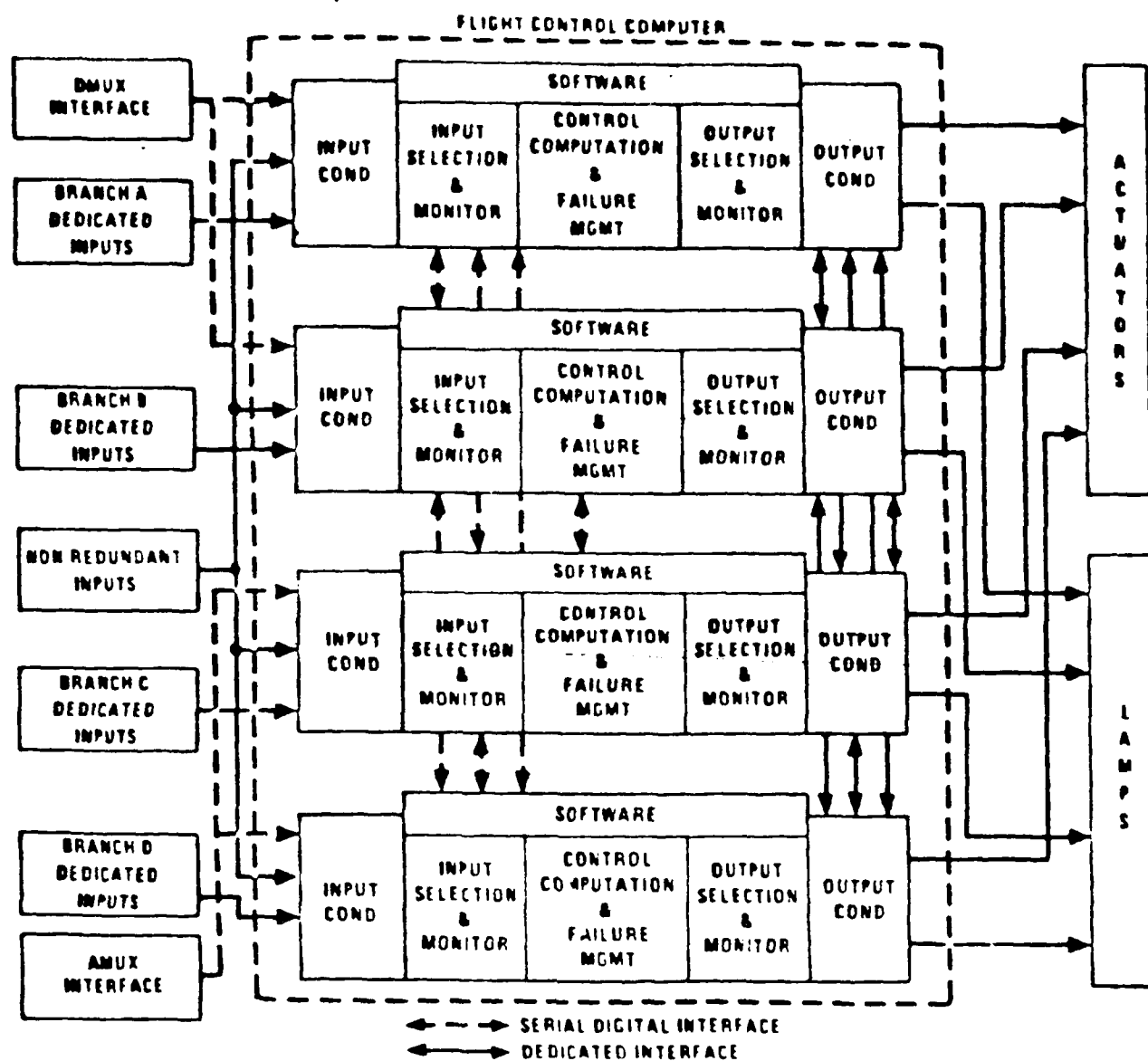


FIGURE D-4 LEADING EDGE FLAP INTERFACE OF THE F-16 DFCS

TABLE D - 3 NATURAL FAULTS IN F-8 DFBW FLIGHTS

No. of Incid.	Component	Mode	Remarks
4	Computer memory failure		Unrecoverable parity errors
2	Computer stopped executing		
1	Interface unit transient fault		Restart restored normal operation
1	Interface unit permanent fault]		

Except where the computer stopped operating the failure was correctly diagnosed by the affected computer as well as by its partners. In failures involving a non-operational computer both partners furnished the correct diagnosis. In none of these 8 failures was there a noticeable control transient, and the aircraft landed routinely on the two channels that remained operational.

In addition to the problems shown above, over 750 simulated faults were induced in the digital portions of the system, and a large number of faults and anomalies were also simulated in the sensors. In all cases the system recovered without incident. The basis for the redundancy management was that all channels were fully operational at take-off, and extensive diagnostics were utilized to validate this assumption. There were no incidents that indicated a deficiency in the diagnostics. Well over 2,000 hours of ground time were accumulated on the equipment. A design deficiency was noted due to the combination of the following:

1. Existence of a latent fault which appears benign when all three channels are operating.
2. A situation in which all three channels do not have an identical image of the system state.
3. Normal reaction to (2) resulting in disablement of more than one channel.

This potential problem was corrected by a software change.

A more recent project at Edwards/Dryden is the Advanced Fighter Technology Integration (AFTI) F-16 program. A total of 118 flight were conducted over a 13 month period ending July 1983. The main objectives of the AFTI program is to investigate aircraft configurations which provide high maneuverability and other operational advantages. Auxiliary control surfaces (canards, wing slats) and non-linear control modes are employed for this purpose. Digital fly-by-wire technology is an essential ingredient of this program. The F-16 installation comprises three identical flight control computers and an analog back-up system. A block diagram of the digital portion is shown in Figure D-5 [MAC83a].

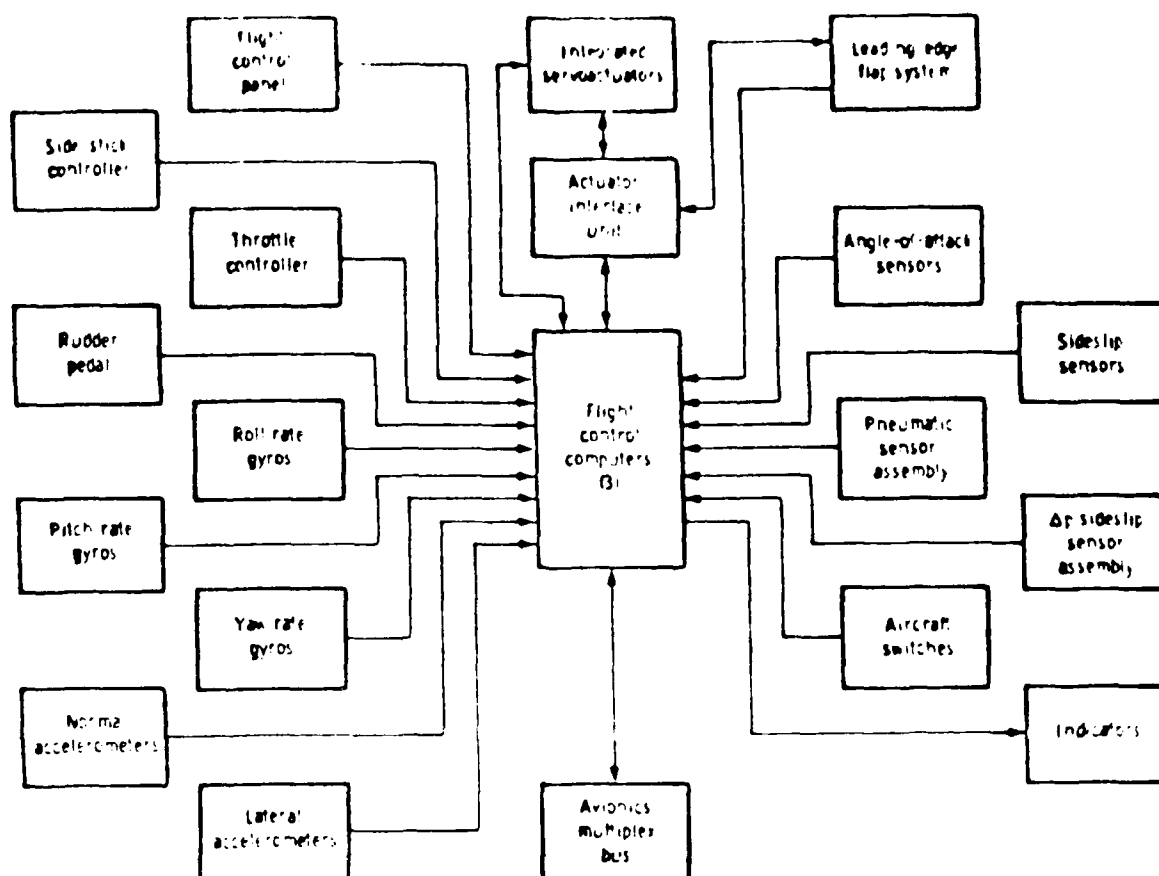


FIGURE D - 5 AFTI F-16 DIGITAL FLIGHT CONTROL SYSTEM

As was the case in the F-8 installation, a number of synchronization and inter-channel communication problems became evident during ground check, and some of these continued into the flight test stage. Small time differences in sampling of analog sensor data were a typical cause of these problems. If the pilot moved the control stick rapidly, the stick position sensed by the three computer channels would differ. Due to the high gain in the control laws, the initially small differences were propagated into large differences at the computer output. The redundancy management program identifies failed computers by looking at the differences in computer outputs. When the differences due to sensor skew exceed a given threshold, one or more computers may be considered to have failed although they are fully operational.

A number of software problems were discovered during ground test and flight test. Over 10 software releases were made to correct these. There were no in-flight computer hardware failures. Perhaps the most serious incident was associated with spurious control mode change commands originating in a cockpit panel. The exact cause of that failure is still unknown. The mode command was received and correctly acted on by all three computers and caused periodic and potentially dangerous actuation of the control surfaces. The pilot perceived these as "rough air" but they were correctly diagnosed on the ground (from telemetry) and corrective action was taken [MACK83b]. A reasonableness check on the frequency and sequence of mode changes can prevent the propagation of single

point failures to flight critical portions of the system.

Both the F-8 and the F-16 programs pointed out the value of extensive ground testing and of pre-flight diagnostics. The fly-by-wire systems were found to be operationally acceptable. The AFTI F-16 program achieved over 20 hours of flight time during several months, and almost 30 hours during the final month. This record of availability in an experimental program is encouraging for the use of digital techniques for flight critical systems.

## Appendix E

### AIRCRAFT ELECTRIC POWER SYSTEMS

Electric power is essential for the operation of most flight critical systems. An important step in the validation of these systems is therefore to analyze the provisions of the power system for supplying essential loads under both normal and abnormal operating conditions. The tendency to reduce cockpit crews makes it desirable or necessary that all power regulation and switching be automatic.

The need for highly reliable electric power will increase in advanced aircraft types because:

- Aerodynamic and propulsion efficiency can be increased by utilization of automatic (electronic) controls which then become essential for safety of flight or for keeping pilot workload at an acceptable level.
- The high cost of maintenance on mechanical, hydraulic, and pneumatic components promotes the introduction of electrical or electronic equivalents. Studies of the replacement of hydraulic actuators by electric ones is an example of this tendency.
- Advantages claimed for the all-electric airplane (a concept that is in the demonstration phase).

In addition to the need for constant availability of electric power, it is required that the power system not serve as either source or conductor of electromagnetic interference (EMI) which can cause unpredictable operating problems in digital equipment. The topics explicitly related to validation are discussed in later headings of this section. An overview of the principal components and design considerations for the electric power system in current aircraft is presented below.



## E.1 AVAILABILITY OF POWER DURING NORMAL FLIGHT CONDITION.

### 1 Aircraft Electric Power Generation

The primary sources of normal electric power on current aircraft are engine mounted generators. Conventionally, the variable speed engine output is converted by a hydromechanical drive, e. g., a constant speed drive (CSD) to a fixed speed of 8,000 or 12,000 RPM at the generator shaft. This enables the generator to provide a constant frequency 400 hertz (400 Hz) output, the customary frequency for aircraft use. The output voltage is usually 3 phase, 115/200 volts AC.

Modern generating systems combine the CSD and generator into a single, oil-cooled unit termed an integrated drive generator (IDG). The most recent units mount the generator and CSD in a side by side rather than an in-line configuration; these are referred to as IDGS (integrated drive generator - side by side). The current state-of-the-art is represented by a high speed, 2-pole generator with an inside-out rotor: the high speed of 24,000 RPM allows significant weight reduction and a smaller diameter rotor allows the higher speed without reducing the reliability.

In most current systems constant speed is maintained by a mechanical or hydraulic governor with an electromagnetic trim head for fine control. Electronic means for obtaining constant frequency output are finding increasing acceptance, and some of these are mentioned later.

The CSD units have experienced oil starvation during zero G and other severe attitude conditions because the intake to the pressurization system ran dry. This caused loss of electric power at a time when there might be other emergencies and has been responsible for a number of serious accidents. For the latest designs it is claimed that control oil pressure is being maintained at all times by means of a prioritized oil system.

Problems with reliability and the overhaul costs of constant speed drives and their derivatives have led to the development of an electronic equivalent, generally referred to as variable speed, constant frequency (VSCF) drives. VSCF units utilize a high speed alternator directly driven by the engine drive pad at a variable speed (12,000 or 27,000 RPM). In the DC-link VSCF system the AC is first converted to DC, then to constant frequency 400 Hertz AC. In the cyclo-converter VSCF the variable frequency AC is converted directly to constant frequency 400 Hertz AC. The cyclo-converter system utilizes SCR's (silicon controlled rectifiers) for the power electronics which must operate at a lower temperature than the transistors and rectifiers used in the DC-link system. The latter approach thus requires less environmental conditioning and it also uses fewer components than the cyclo-converter system. Both systems require a high speed engine pad.

The cyclo-converter VSCF system has been installed on the A-1 and F-19 military aircraft. The DC-link VSCF system has been utilized on the AV-8B, F-20 and the Gulf Stream III; in addition, a DC-link VSCF unit has been installed as a back up system on the F-16. VSCF systems promise better power quality, higher reliability and, above all, reduced life cycle costs since overhaul and repair of power conversion electronic devices is less costly than that of hydro-mechanical units.

Two advanced generation concepts under study are high voltage DC systems (700 VDC) and hybrid systems which provide wild frequency (directly related to engine rpm) for some loads with conversion equipment for loads not capable of accepting wild frequency. The large number of airborne and ground facilities that require the current standard voltages and frequency pose a serious obstacle to the introduction of new power generation techniques regardless of their intrinsic technical merits.

Voltage and frequency parameters for AC and DC power are governed by MIL-STD-704. However, airframe manufacturers frequently generate their own specifications for demonstrating compliance with MIL-STD-704 or for tailoring it to a specific environment. Special Committee 123 of the Radio Technical Commission for Aeronautics (RTCA) has generated that prescribes test procedures of airborne electronic and electrical equipment, and which is widely accepted in the industry.

A new military specification, MIL-E-23001, has been written for cyclo-converter VSCF systems. Most provisions of that document can also be used for DC-link VSCF systems.

## 2 Generating System Reliability and Availability

The major factor in assuring power system reliability and availability is redundancy of the power sources. When a single engine, single primary generator design is employed, redundancy is provided by an emergency power source not dependent on engine drive power. Figure E-1 shows a typical electrical schematic of such a system. For multi-engine, multi-generator designs, a number of system architectures are in use. When two generators are employed, the system is usually designed for isolated operation with cross-tie capability; this provides complete dual redundancy. In primary electrical systems, each generator and its associated controls, relays and bus is considered a channel. In a two-channel isolated system, each electrical channel is separated during normal operation but the two channels can be connected through the cross tie relay.

Figure E-2 shows a typical two-channel isolated system. In these isolated systems the nature of the failure determines whether or not the system will allow the channels to be cross-tied. If the fault is in the portion of the system isolated by the generator relay (GR) opening, the cross tie is allowed to close. The cross tie is locked out if the generator relay is not open because then there is a high probability of a fault in the load side and closing of the cross-tie could cause the loss of the second generating channel. Lockouts can be removed and a reset attempted by manual action but this involves some added risk. The concepts shown in Figure E-2 can be extended to configurations involving more than two engines.

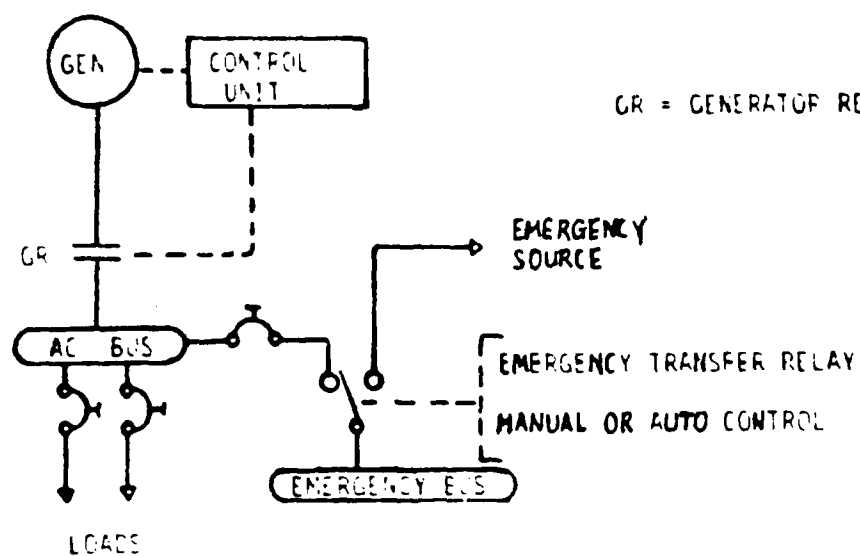


FIGURE E - 1 TYPICAL SINGLE GENERATOR INSTALLATION

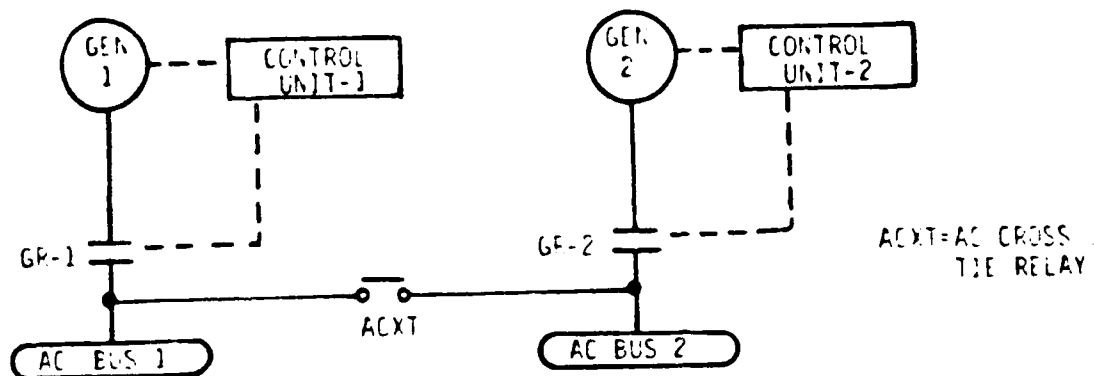


FIGURE E - 2 TYPICAL TWO GENERATOR INSTALLATION

### 3 Control and Protection

The generator control unit provided for each channel contains the required control and protection features. Controls include the voltage regulator which keeps the voltage within the required limits and, in paralleled systems, also means for load control (equal distribution of loads). Control functions are backed up by protective functions which take the channel off the line when the control fails. Protective features on isolated systems include:

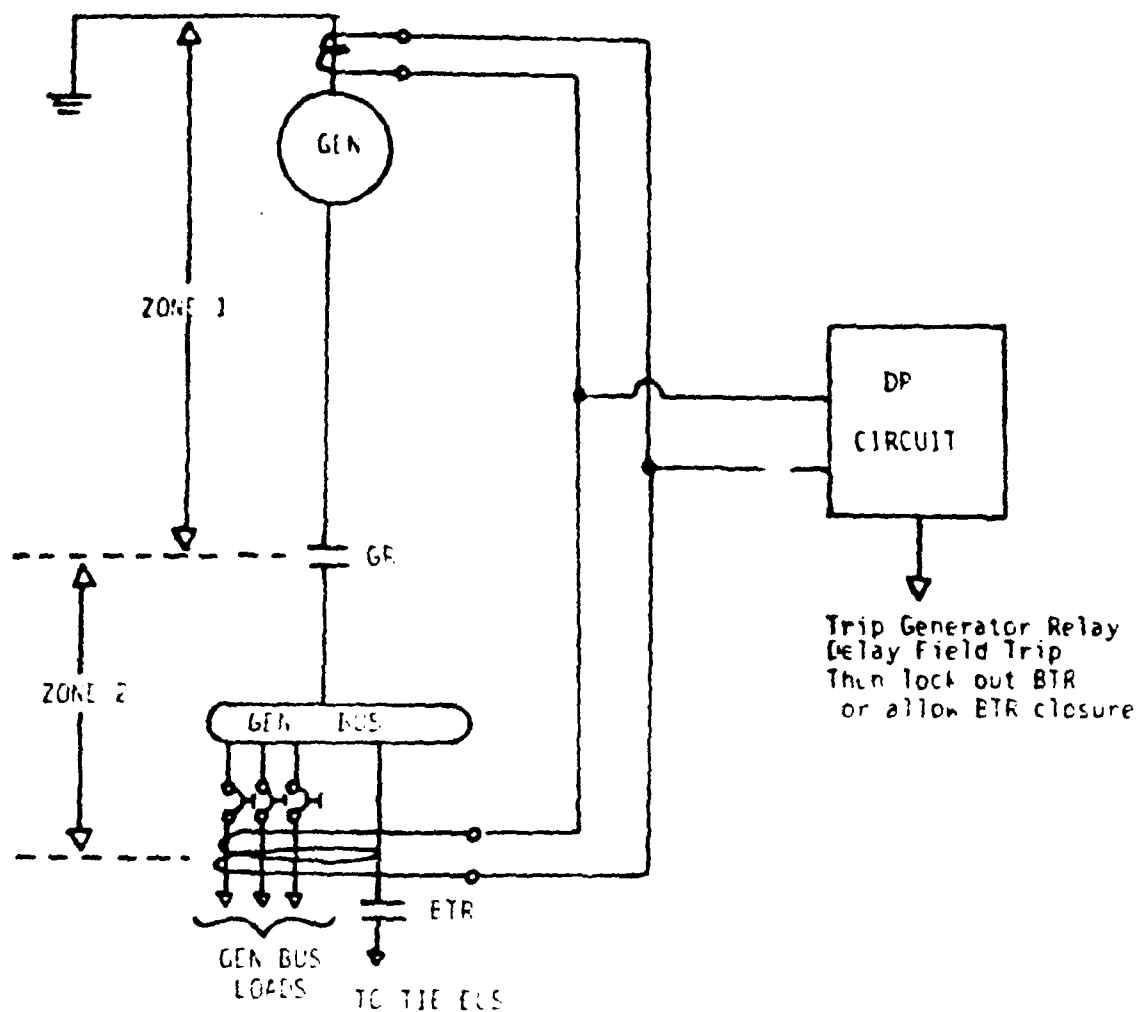
- over/under voltage,
- over/under frequency,
- open phase,
- phase rotation,
- overcurrent,
- ground faults, and
- phase-to-phase faults.

Paralleled systems add over/under excitation and unbalanced current. Frequency control is provided by the constant speed drive portion of IDG's or by the electronics on VSCF systems. Generally, control units on all channels are identical and can be interchanged to allow good channels with failed control units to be kept operational.

Detection for protective features is provided by current transformers and sense leads. Differential current transformers are used to detect fault currents and determine what zone they are in. An example of the use of differential current transformers is shown in Figure E-3.

Control and protection power is generated from permanent magnet generators mounted on the main generator shaft. Thus, the control and protection of the electrical system is not affected by faults in the generator that it controls.

The latest control innovations incorporate momentary paralleling on bus transfers to provide interrupt-free power for normal operation; this reducing transient effects, especially on start ups and shut downs. Automatic control and protection is being provided to a high degree to reduce crew work loads (especially during critical phases of flight) and to allow crew operation by two individuals. Finally, for most operation and failures no crew action is required.



#### WORD LOGIC

1. Sense a differential current (40 amps or more)-Open BTR
2. Open CB but retain field excitation
3. Sense that differential current remains--indicates Zone 1 fault--trip field relay then allow BTR to close
4. Sense that differential current is gone--indicates Zone 2 fault--trip field relay and lock out BTR

FIGURE E - 3 DIFFERENTIAL CURRENT AND ZONE PROTECTION

## E.2 Power Conversion

Power conversion in aircraft may take two forms:

- Converting primary DC power to 400 hertz AC.
- Converting primary AC power to DC.

The former is usually found in older aircraft while the latter is the practice in all recent aircraft designs. Conversion to AC is accomplished in solid state inverters that provide 26 V or 115 V 400 Hz single phase output. Where three phase AC is required, three inverters are connected to a common control unit that provides the proper phase relationship. For aircraft that generate primary AC power the required DC is usually provided by transformer rectifiers. The latter may use multiple windings to create a 12 to 24 phase output which reduces the filtering required for an acceptably low ripple in the output. Circuit breakers are used for protection at the input and output of the conversion units. Where conversion units are operated in parallel, isolating diodes may be used at the output for additional protection. The standard DC voltage presently utilized is 28 volts. The AC voltage input to the transformer rectifier is regulated and this is usually sufficient for stability of the DC output.

Many aircraft require 28 V AC single phase for lighting and instruments. This is usually derived from the 115 V AC by autotransformers which supply local 28 VAC buses.

Some aircraft provide 60 Hz AC power to facilitate use of ordinary appliances. In specialized uses (e. g., for life support equipment in medical evacuation aircraft) this power may become critical and will require validation. It is presently furnished by solid state power conversion equipment which uses a three phase, 400 Hertz, 115/200 volt AC input and supplies single phase, 115 volt 60 Hz output.

Existing specifications require that non-standard voltages be generated internal to the using equipment. Thus, digital components that use regulated  $\pm 15$  volts or  $\pm 5$  volts power must contain internal power converters. As the uses of digital equipment on board aircraft increase some changes in this policy may be expected.

## 2.1 Non-Flight and Emergency Conditions

### 1 External Power

Virtually all aircraft use a standard external power receptacle which allows an external source supply electrical power to the aircraft. The external power is usually connected to the AC tie bus via the external power relay. From there the bus tie relays can be used to apply the power to any or all of the generator buses. External power protection usually consist of:

- over/undervoltage,
- over/underfrequency, and
- open phase and incorrect phase rotation.

The components for implementing this protection are located in the external power panel. Lights are provided at the panel to indicate when the external power quality is acceptable and when external power is not in use so that the plug can be safely removed. Should external power quality be unacceptable for aircraft use, the external power relay is tripped, disconnecting this source from the aircraft buses. The use of external power for all ground functions is preferred since this power is considerably less expensive than running an engine or the APU (see below).

### 2 Auxiliary Power

Auxiliary power units (APUs) are installed in many aircraft to provide an autonomous source of ground power, particularly for starting engines in locations where external power is not available. The auxiliary power unit consists of an engine driven alternator, usually with good speed regulation so that no additional means of frequency control are necessary. This generator is usually connected to the AC tie bus via an auxiliary power relay, using the existing bus tie relays to apply this power to any or all of the generator buses. Automatic overcurrent protection APU generators is considered essential since the cockpit is not always attended during APU operation.

In typical operation the aircraft battery is used to start the APU, and the current supplied by the latter is then used to start the aircraft engines. APUs are designed for ground operation and are usually not a source of in-flight emergency power (see below).

### 3 Availability of Power under Emergency Conditions

Emergency power requirements fall into two categories. The first is an all engine out condition. This requires emergency power capability to control the flight so that an engine restart profile can be followed and in the case of failure to restart engines allows enough capability for a controlled landing. Requirements in this instance are for no more than 1/2 hour of flight. The second category is an all generator out condition in which engine power is available. For this condition and an overwater flight emergency, power could be required for 2 to 3 hours.



Under emergency conditions only loads connected to the AC and DC emergency buses are powered. Such loads include communications, flight and engine standby instruments, all necessary flight controls, minimal navigation equipment, required indications, ignition and some cockpit lighting. Since loss of power could occur in critical phases of flight, transfers to emergency power should be automatic. Manual transfer capability is provided as a backup to the automatic controls and to allow testing.

The following sources of emergency power are in current use:

- Storage batteries -- frequently the same battery used for autonomous engine starting
- Ram air turbine -- this may drive an emergency hydraulic power supply which in turn drives an alternator or combination AC and DC generator
- Air driven generator -- usually manually deployed (via cables) into the airstream.

The main disadvantage of batteries is their limited capacity and hence the resulting restriction on the duration of flight which they can support. Both of the other sources can support virtually unlimited flight duration if there is propulsive power available, and this is the condition under which prolonged flight on emergency power is required. Some recent military aircraft employ a monofuel powered emergency generator.

Testing the capability of the emergency power system is an item of major concern. Where a battery system is used for emergency power and APU starting, the APU starting is a good test of battery availability. Additionally, emergency power can be selected and the system can be fully tested on the ground and in the air. The ram air turbine can be tested by use of an auxiliary hydraulic supply to drive the generator. The air driven generator cannot be directly tested on the ground; it is customary to connect an auxiliary hydraulic motor to an extension of the generator shaft to provide the driving force.

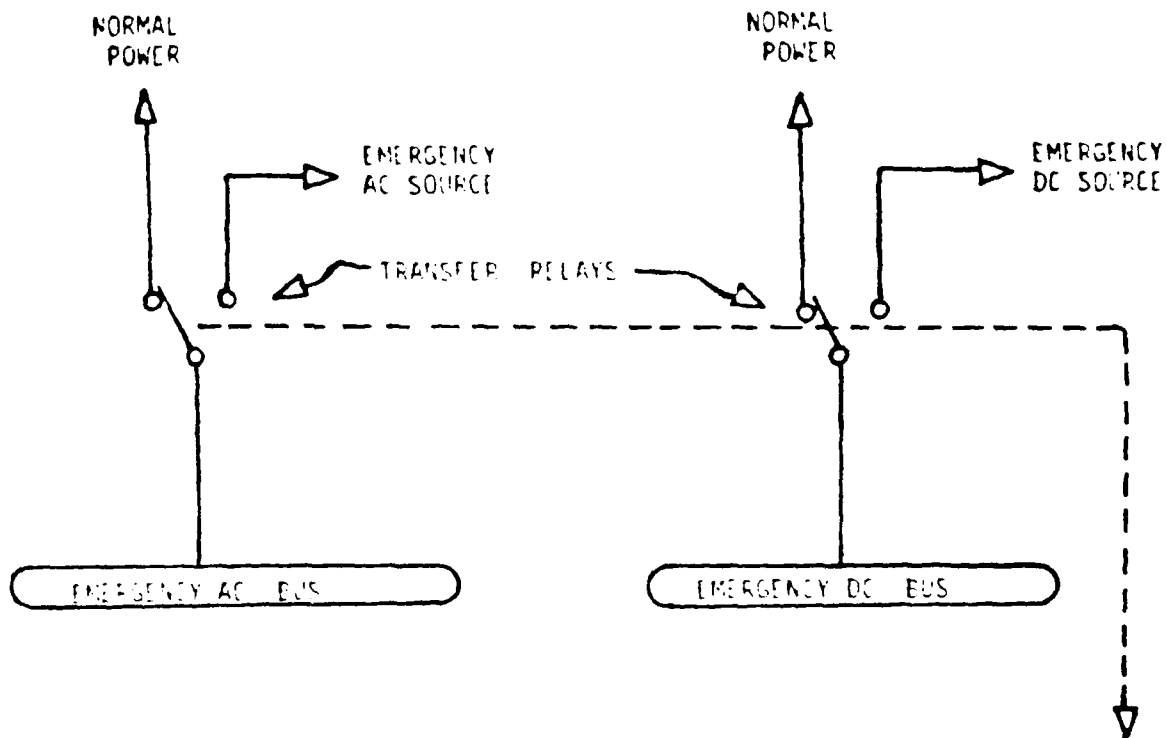
A typical emergency power load transfer arrangement is shown in Figure E-4.

#### E.4 Built-In Test (BIT)

Extensive pre-flight testing of the electric power system is impractical because of:

- the time required for a comprehensive test of all functions,
- the expense of running the engines (required to test all generators), and
- inability to simulate some loads which are active only in flight (engine inlet de-icers, etc.).

For these reasons, built-in test capability has been provided to capture the occurrence and environment of all abnormal (and some normal) conditions of the



Transfers when either emergency bus is sensed as being deactivated in flight. Requires manual selection of control switch to off to allow transfer back to normal power (prevents cycling).

FIGURE E - 4 EMERGENCY LOAD TRANSFER

electric power system that arise during routine operation. The occurrence of such conditions is indicated to the crew, and more detailed data are made available for maintenance purposes.

Most recent aircraft provide three distinct BIT functions:

- Continuous BIT -- used during normal operation, start up and shut down to automatically detect, store and indicate a failure condition and upon interrogation indicate the type of failure and the corrective action required.
- Periodic BIT -- providing the capability to periodically check the passive protective features, such as circuit breakers, phase rotation sensors, and ground fault detectors.
- Stored Memory BIT -- used to store selected responses to test inputs, primarily for use in off-line and shop testing.

The latter capability is particularly valuable for tracking down transient failure symptoms.

In present installations the BIT is a dedicated function serving only the electric power system. Controls for the BIT and indications from it are associated with the electric power panels in the cockpit. There are tendencies to integrate BIT functions from several aircraft systems in order to reduce the number of cockpit controls and to furnish improved readout capabilities (e. g., a single cathode ray display for all maintenance requirements).

#### E.5 Control of Electromagnetic Interference (EMI)

The increasing use of digital components aboard aircraft has led to heightened concern with the reduction of EMI or, in a positive sense, with improving the electromagnetic compatibility of all aircraft systems. Digital components are particularly vulnerable because:

- they operate at low voltages,
- their extremely fast response time, and
- many digital functions include latches so that once an improper state has been induced the function will not return to the normal state if the stimulus (the EMI) ceases.

## 1 EMI Contributions of the Power System

The primary means of generating interference are conduction and radiation, and both of these may be caused by the electric power system. Conducted interference is generated by:

- processes within the generation and control system, e. g., harmonics of the generator output voltage,
- switching of buses or of individual heavy or inductive loads,
- "pick-up" by conductors leading to the digital equipment, and
- voltage differences in the ground plane.

Permissible manifestations of the first two mechanisms are defined in MIL-STD-704. Using equipment must be capable of tolerating these, and the most common techniques for accomplishing this are line filters and voltage regulators. However, external power supplies do not always conform to MIL-STD-704, and there have been a number of incidents in which faulty operation or even permanent damage to sensitive equipment has been caused by ground power supplies which were outside the limits of the standard. Transients associated with switching from one supply to another have also been an occasional cause of EMI related failures. The employment of make-before-break contacts on transfer relays, which is now coming into practice, is expected to alleviate these problems.

Pick-up can be reduced by installation practices which include use of twisted and/or shielded conductors and bundling of cables, based on their voltage and current characteristics. Typical groups for this purpose are:

- Power conductors to heavy electrical loads.
- Power conductors for instrumentation and electronic equipment.
- Extremely susceptible lines (pyrotechnics, antenna wiring).
- Sensitive audio, video, synchro, and digital data lines.
- Equipment connections (which may constitute a combination of the above) limited to short lengths (about 3 feet).

Bundles of different conductor types are separated by 3 to 12 inches, depending on the length of the run and the relative susceptibility of the signals carried.

Radiated interference can be caused within the electric power system by any arcing, such as is associated with opening of a contact in circuits serving inductive loads, by chattering relay contacts, and by fields set up by high voltages in using equipment. Because there are other sources of radiated EMI, such as triboelectric and atmospheric discharges, the primary preventive measure is the shielding of all sensitive circuits.

END

Dtic

7-86